National Library
of Canada

Canadian Theses Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Service des thèses canadiennes

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-52799-4

Canada

THE UNIVERSITY OF ALBERTA

MODELS AND METHODS FOR RECURSIVE IDENTIFICATION

BY

©

MICHAEL DAVID ROGERS

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

SPRING 1989

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: MICHAEL DAVID ROGERS
TITLE OF THESIS: MODELS AND METHODS FOR RECURSIVE IDENTIFICATION
DEGREE: MASTER OF SCIENCE
YEAR THIS DEGREE GRANTED: 1989

Permission is hereby granted to THE UNIVERSITY OF ALBERTA to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

Mich Rogers

71 Denault
Kirkland, Quebec
H9J 3J2

Date: January 25, 1989

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled MODELS AND METHODS FOR RECURSIVE IDENTIFICATION submitted by MICHAEL DAVID ROGERS in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in PROCESS CONTROL.

Dr. S.L. Shah
Co-Supervisor

Dr. R.K. Wood
Co-Supervisor

Dr. P.M. Huck

Date:

To my wife Kathryn, without whose endless patience, love and understanding this work would have never reached completion.

Abstract

The use of adaptive control in industry has been limited due to problems in the theory and implementation of recursive identification algorithms. This work addresses a few of the problems involved with model structures, algorithms, data forgetting and bounded noise descriptions.

Guidelines for choosing a model structure are presented from a survey of recent literature of linear models and recursive parameter estimation methods.

Problems with modeling the steady state displacement by differencing the data are examined. The C polynomial allows the disturbance to be described as a combination of Brownian motion and white noise hence the value of the C parameters may be related to some physical interpretation of the system. The C polynomial is equivalent to a Kalman filter for estimating the displacement term μ.

A new exponential forgetting factor which maintains the determinant of the P matrix equal to a constant is presented. It is compared to others using both a simulated example and an application to closed loop data from a pilot scale distillation column. This approach is robust under noisy, and non-persistently excited conditions.

The directional forgetting algorithm of Kulhavy and Karny, (1984) is applied to the simulation example and distillation column data. This approach has several advantages over exponential forgetting. The algorithm is inherently robust, and the value of the forgetting factor is not related to the dimension of the problem.

The effect of the width of a prediction error dead zone on the estimation accuracy is evaluated using the distillation column data. Smaller

dead zones provided a diminishing return on accuracy at the expense of dramatically increased computation load.

The new variable dead zone algorithm of Dasgupta and Huang (1987) was also applied to the distillation column data. Large initial variations in the parameters and lack of tolerance to underestimation of the noise bound exclude the use of this algorithm for process applications.

Acknowledgements

The author wishes to express his sincere gratitude to Dr. S.L. Shah, and Dr. R.K. Wood for their supervision, help and encouragement during the course of this research.

I would like to give a special thanks to my office mate and friend David Shook, with whom I enjoyed many conversations and good times.

Thanks are due to the technical staff of the DACS centre, Instrument Shop and Machine Shop for their help with the distillation column equipment.

This thesis was produced in its entirety on an IBM PC-AT computer using the CHIWRITER word processor.

The author would like to thank the Department of Chemical Engineering and the Natural Sciences and Engineering Research Council of Canada for their financial support.

Table of Contents

Abstract v
Acknowledgements vi
List of Tables xii
List of Figures xiii
Nomenclature xvi

Chapter 1 Introduction

1.1 Introduction to Recusive Identification 1
1.2 Organization of the Thesis 4

Chapter 2 Process Models for Self Tuning Control

2.1 Introduction 7
2.2 The Deterministic Dynamic Model 8
2.3 Disturbances 9
2.4 Steady State Information and Low Frequency Disturbances 14
2.5 Methods of Pre-Treating the Data 15
    2.5.1 Deviations from Physical Equilibrium 16
    2.5.2 The Incremental Approach 16
    2.5.3 A Mean Deviational Approach 17
2.6 Including the Displacement In the Disturbance Model 18
    2.6.1 The Positional Model 19
    2.6.2 A Noise Model with Integration 20
2.7 A Family of Transfer Function Models 22
2.8 The Equation Error Description 22
    2.8.1 The ARX Form 23
    2.8.2 The ARMAX Model Structure 25
    2.8.3 Other Equation Error Forms 25
2.9 The Output Error Model Structure 26
    2.9.1 The Measurement Noise Structure 28
    2.9.2 The Box-Jenkins Model Stucture 29
2.10 The General Model Structure 30
2.11 Summary 31

Chapter 3 **Recursive Identification and Adaptive Filtering**

3.1 Introduction 36
3.2 The Prediction Error Methods in Identification 37
3.3 False Local Minima 40
3.4 Recursive Prediction Error Methods 41
3.5 The Projection Algorithm 41
3.6 The Method of Least Squares 43
3.7 Asymptotic Properties of the RLS Algorithm 46
3.8 The Generalized Recursive Maximum Likelihood Method 51
3.9 Choice of the Gradient for the General Model Structure 54
3.10 *A Posteriori* Variables 58
3.11 The Pseudo-Linear Regression 59
3.12 Convergence and Asymptotic Properties of the RML Method 62
3.13 Transient Behavior of the RML Algorithm 63
3.14 Convergence and Asymptotic Results of the PLR Method 64
3.15 Generalised Least Squares 67
3.16 The Instrumental Variables Method 70
3.17 Choice of the Instrumental Vector 73
3.18 Summary 78

Chapter 4 **Application of the Differencing Operator**

4.1 Introduction 81
4.2 The Integrating or Brownian Motion Noise Model 82
4.3 Frequency Response Properties of the Differencing Operator 86
4.4 Interpretation of the T Filter 91
4.5 Kalman Filter Interpretaion of the ARIMAX Structure 94
4.6 Summary 104

Chapter 5 **Exponential Forgetting Factors**

5.1 Introduction 106
5.2 Estimation Shut-off In RLS Methods 108
5.3 A Geometric Interpretation of the P Matrix 109
5.4 Exponential Forgetting Factors 112
5.5 Constant Information Forgetting Factor 116

5.6 Constant Trace Forgetting Factor 120
5.7 Constant Determinant Forgetting Factor 121
5.8 Rank Deficient Data 123
5.9 Simulation Example 125
5.10 Experimental Evaluation 132
5.11 Experimental Observations 143
5.12 Conclusions 144
5.13 Summary 144

Chapter 6 **Directional Forgetting**
6.1 Introduction 147
6.2 A Bayesian Approach to Identification 148
6.3 Directional Forgetting Applied to the Least Squares Method 151
6.4 Choice of the Forgetting Factor 153
6.5 Interpretations of the Directional Forgetting Method 156
6.6 Reduced Rank Data 158
6.7 Simulation Example 159
6.8 Experimental Evaluation 162
6.9 Summary 165

Chapter 7 **Application of Least Squares Type Methods to Bounded Disturbances**
7.1 Introduction 167
7.2 Systems with Bounded Disturbances 168
7.3 Prediction Error Dead Zone 169
7.4 Experimental Evaluation of the Dead Zone Modification 171
7.5 A Set Theoretic Approach to Parameter Estimation 174
7.6 The Modified Least Squares Method 178
7.7 Experimental Evaluation 183
7.8 Observations on the MLS Algorithm 187
7.9 Summary 188

Chapter 8 **Conclusions and Recomendations**
8.1 Conclusions 190
8.2 Recommendations for Future Work 194

References 197

Appendix A Proof of Theorem 5.1 204
Appendix B Computer Programs 210

List of Tables

2.1 Some Common Special Cases of the General Model Structure 31

4.1 Experimental Conditions for Simulation Example 98

4.2 Comparison of Transfer Function Estimation versus Kalman Filtering Estimation of the Closure term 101

5.1 Summary of Final Conditions of the P Matrix 131

List of Figures

2.1 The Equation Error Model Form 23
2.2 The Output Error Model Form 27
4.1 Equivalent Representations of the ARIMAX structure 83
4.2a Bode Plot of the Differencing Filter 88
4.2b Cumulative Spectral Density of Differenced White Noise 89
4.3 Bode Plot of the Mean Deviational Filter 93
4.4a Parameter Trajectories Run #1 102
4.4b Parameter Trajectories Run #2 102
4.4c Parameter Trajectories Run #3 103
5.1 Confidence Bounds in R^2 111
5.2b Effect of Exponential Forgetting on the Confidence Bound Ellipse 115
5.3a Parameter/Forgetting Factor Trajectories with the Constant Information Forgetting Factor 127
5.3b Parameter/Forgetting Factor Trajectories Constant Trace Forgetting Factor 128
5.3c Parameter/Forgetting Factor Trajectories with Constant Determinant Forgetting Factor 128
5.4a Covariance Bounds - Constant Information Forgetting Factor 130
5.4b Covariance Bounds - with Constant Trace Forgetting Factor 130
5.4c Covariance Bounds - with Constant Determinant Forgetting Factor 131
5.5 Schematic Diagram of the Pilot Scale Distillation Column 133
5.6 Distillation Column Data 134
5.7a Parameter Trajectories - Constant Information Forgetting Factor Applied to Distillation Column Data 137
5.7b Forgetting Factor - Constant Information Forgetting Factor Applied to Distillation Column Data 138
5.7c Matrix Determinant - Constant Information Forgetting Factor Applied to Distillation Column Data 138
5.7d Condition Number - Constant Information Forgetting Factor Applied to Distillation Column Data 138

5.8a Parameter Trajectories - Constant Trace Forgetting Factor Applied to Distillation Column Data 139
5.8b Forgetting Factor - Constant Trace Forgetting Factor Applied to Distillation Column Data 140
5.8c Matrix Determinant - Constant Trace Forgetting Factor Applied to Distillation Column Data 140
5.8d Condition Number - Constant Trace Forgetting Factor Applied to Distillation Column Data 140
5.9a Parameter Trajectories - Constant Determinant Forgetting Factor Applied to Distillation Column Data 141
5.9b Forgetting Factor - Constant Determinant Forgetting Factor Applied to Distillation Column Data 142
5.9c Matrix Determinant - Constant Determinant Forgetting Factor Applied to Distillation Column Data 142
5.9d Condition Number - Constant Determinant Forgetting Factor Applied to Distillation Column Data 142
6.1 Effect of Directional Forgetting on the Confidence Bound Ellipse 157
6.2a Parameter trajectory for the directional forgetting algorithm applied to the simulation example 160
6.2b Forgetting Factor trajectory for the directional forgetting algorithm applied to the simulation example 160
6.2c Data Weighting trajectory for the directional forgetting algorithm applied to the simulation example 160
6.3 Evolution of Confidence Bound Ellipses - Directional forgetting algorithm applied to the simulation example 161
6.4a Parameter Trajectories - Directional forgetting applied to the distillation column data 163
6.4b Forgetting Factors - Directional forgetting applied to the distillation column data 164
6.4c Weighting Factors - Directional forgetting applied to the distillation column data 164
6.4d Determinant of P Matrix - Directional forgetting applied to the distillation column data 164
6.4e Condition Number - Directional forgetting applied to

to the distillation column data 165

7.1a Sum of Prediction Errors Squared versus Width of Dead Zone 171

7.1b Fraction of Data Used versus Width of Dead Zone 172

7.2 Intersection of $\Theta(t-1)$ with $S(t)$ 178

7.3 Effect of β Weighting on the Bounding Ellipse 180

7.4a Parameter Estimates Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.07$ 184

7.4b Parameter Estimates Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.04$ 184

7.4c Parameter Estimates Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.03$ 185

7.5a Weighting Factors Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.07$ 185

7.5b Weighting Factors Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.04$ 186

7.5c Weighting Factors Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.03$ 186

7.6a Lyapunov Bound Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.07$ 186

7.6b Lyapunov Bound Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.04$ 187

7.6c Lyapunov Bound Modified Least Squares applied to Distillation Column Data $\gamma^2 = 0.03$ 187

Nomenclature

Symbols

$A(q^{-1})$	Polynomial in the backshift operator
$B(q^{-1})$	Polynomial in the backshift operator
$C(q^{-1})$	Polynomial in the backshift operator
$D(q^{-1})$	Polynomial in the backshift operator
d	Delay including sampling delay
$E\{\}$	Expected value operator
$F(q^{-1})$	Polynomial in the backshift operator
$G(q^{-1})$	Polynomial in the backshift operator
$G(t)$	Scalar term $\phi^T(t)P(t)\phi(t)$
$H(q^{-1})$	Polynomial in the backshift operator
I	Identity matrix
i	The square root of -1
$K(t)$	Kalman gain vector
$L(q^{-1})$	Polynomial in the backshift operator
$M(q^{-1})$	Polynomial in the backshift operator
n	Dimension of the parameter or regressor vector
$n_a, n_b \ldots$	Order or degree of a polynomial
$P(t)$	Covariance matrix of the parameter estimates
p	Probability density function
q^{-1}	Backshift operator ie. $x(t-1) = q^{-1}x(t)$
$R(t)$	Data matrix
$S(t)$	Region in R^n defined by a degenerate ellipsoid given by data available at time t.ble at time t.
$T(q^{-1})$	Polynomial in the backshift operator

$u(t)$	Process or model input signal
$u_m(t)$	Process input signal in measurement units
$\underline{u}(t)$	Estimate of the steady state or nominal value of the input signal
V_t	The value of a cost function at time t in an off-line analysis
$V(t)$	The value of the cost function at time t calculated recursively.
$v(t)$	Unmeasured disturbance signal
$w(t)$	Undisturbed model output signal
$x(t)$	State variable
$y(t)$	Process or model output signal
$y_m(t)$	Process output signal in measurement units
$\underline{y}$	Estimate of the steady state or nominal value of output signal.
y_{ss}	Steady state or nominal value of the output signal
z	The Z-transform operator, ie. $z=e^{i\omega}$
$\alpha(t)$	Data weighting sequence
$\beta(t)$	Weighting factor
χ^2	Chi-Squared probability distribution
$\delta(t)$	Pulse function at time t
δ_1,δ_2	Dead-zone criterion
ϵ	Prediction error
ϕ	The regressor or data vector
γ	Noise bound
ψ	The gradient vector
λ	Forgetting factor

μ	Closure or displacement term
θ	Parameter vector
θ_0	Parameter vector which describes the true system
$\hat{\theta}(t)$	Recursively calculated estimate of the parameters
θ^*	Convergence point of the parameter estimate
$\tilde{\theta}(t)$	Parameter error vector $\theta_0-\hat{\theta}(t)$
σ	Standard deviation
τ_p	Time constant of a process
ω	Frequency variable (radians/s)
$\xi(t)$	Discrete white noise signal
ζ	Instrumental variable
Δ	Differencing operator $\equiv 1-q^{-1}$
Λ	Region enclosed in Σ^n
Θ	Membership set of the parameter estimates
$\Omega(\omega)$	Spectral density as a function of frequency

Chapter 1

Introduction

1.1 Introduction to Recursive Identification

Modern control systems in use today frequently rely on linear dynamic models of a system to describe transient behavior. A major portion of the work involved in designing a control system is the development of these models. In general, there are two approaches that may be taken to obtain the model. If the physical system is well known and there is sufficient mathematical theory available to describe all of its internal mechanisms then the dynamic model may be written as a set of differential equations, whose structure and constants correspond directly to physical relationships and quantities. For many realistic situations, however this approach is impractical because of the lack of sufficient knowledge of the system, or the complexity of the resulting differential equations. A more direct and often more practical approach is made by analysis of dynamic data gathered from the system. This approach is known as model identification. Typically the steps involved in model identification may be classified in the following general categories:

1. Design of experimental conditions
2. Collecting suitable data
3. Choosing the model structure
4. Estimating the parameters of the model
5. Verifying the resulting model choice

In off-line or non-recursive model identification each of the above steps is well defined. All of the steps, perhaps with the exclusion of step 2, may be carried out independent of the operation of the system and hence any reasonable amount of time and effort may be taken to ensure that the resulting model is valid and appropriate. In some instances, however, it is found that decisions relating to the dynamic behavior of the process must be made in a changing environment; one in which the dynamic model derived from a previous range of operating conditions is, for some reason, invalid or too inaccurate to be applied to the present conditions. This situation motivates the use of recursive identification, in which the model is continually modified with each new piece of information to improve its accuracy and to track any time varying or non-linear behavior in the system. Such an approach attempts to ensure that the model is derived from current data, and therefore is the most appropriate under the present operating conditions. Clearly this approach to identification cannot be easily dissected in terms of the five steps previously defined, since all of these must be occurring, in some form, each time the model is updated. Since the identification of the model must be concurrent with the operation of the process, the decisions relating to the model identification steps must be made in a limited time frame, with limited information available, and possibly limited computing power.

Recursive identification of a dynamic model is a cornerstone of any model based adaptive controller. The success or failure of an adaptive controller in practice will be highly dependent on how well the recursive identification performs. Indeed much of the reluctance to the adoption of adaptive control in industry is due to problems and unfamiliar theory associated with recursive identification (Tjokro, 1984). Numerous recent studies have focused on improving the robustness of the control law (Cluett, 1987, Song *et al.* 1986) in order to reduce the reliance on model accuracy, however these are often made at the direct expense of performance. Heuristically speaking, model insensitivity (robustness) is necessary to provide global stability and hence reliability of the control. However it is argued here that the accuracy and reliability of the model, and consequently the recursive identification scheme are essential to improve the performance of a model based adaptive control system. This statement is the main motivation behind the present work, although the study of adaptive control per se will not be pursued. The present work was undertaken to examine some of the problems in recursive identification which have inhibited the acceptance of adaptive control, to critically review the existing methods available in handling those problems, and to suggest new solutions and approaches. A major effort in the work was in providing understandable interpretations of many of the recent innovations in the field.

The major distinctions of this work are:

1. A unified approach to the choice of model structure is presented which includes the problem of dealing with the steady state levels.

2. A new interpretation of the C polynomial of the ARIMAX model structure is proposed. This observes that the root locations indicate the proportion of stationary noise to a Brownian-motion disturbance, and hence the value of the C parameters may be related to some physical interpretation of the system.

3. A new forgetting factor which maintains the determinant of P at a user specified value. Such a choice will inherently avoid matrix singularity, and from a statistical point of view, is a more meaningful

measure of the P matrix than the trace. When the U-D factorisation method of Bierman (1976) is employed, det(P), unlike tr(P), is easily computed and monitored without requiring re-construction of the P-matrix at each interval. The constant determinant method is shown to be robust when applied to rank deficient data in the presence of noise.

4. The directional forgetting algorithm of Kulhavy and Karny (1984) is shown to be an inherently robust approach when applied to rank deficient data.

5. The effect of applying a prediction error dead zone criterion on the identification step is examined by application to an off line data set from a pilot scale distillation column. It is observed that larger dead zones increase the sensitivity of the estimates to outliers.

1.2 Organisation of the Thesis

The basic approaches to constructing a linear model of a process are reviewed in Chapter 2. Various forms of the resulting model structure are presented with recommendations for their application. A discussion of the problem of including the steady-state information in a dynamic model is presented. Various commonly employed approaches to the problem are presented and critically discussed.

The output error and equation error forms of the linear dynamic model and methods for treating non-zero equilibrium levels are unified by the consideration of a general model structure from which any of these may be subsumed. Recommendations are made for choosing a particular model within the framework defined.

In Chapter 3 a review of the recursive algorithms for estimating the parameters of the general model structure is presented. The algorithms are interpreted within a general recursive prediction error method framework. Ordinary recursive least squares, recursive maximum likelihood, extended least squares, Landau's model reference output error method, generalised least squares and instrumental variable methods are presented and reviewed. Recommendations for the use of these methods based on a review of the literature are provided.

An examination of the use of the "Brownian" motion model as a method of describing the steady state closure term is the subject of Chapter 4. The inherent problems with this approach are examined and methods of overcoming them are presented. The ARIMAX model structure is interpreted as a linear combination of Brownian and stationary noise signals. This new interpretation sheds some understanding of how the root locations of the C polynomial are related to the structure of the noise and disturbance processes. An interpretation of the ARIMAX model as an "optimal" or Kalman filtering approach to estimating the displacement or closure term is explained and demonstrated with a simulated example.

Chapter 5 examines various approaches to the idea of exponential data forgetting in the application of ordinary least squares. The variable forgetting factor is interpreted as a way of incorporating the advantages of the "constant gain" projection algorithm methods to the "least-squares type" methods which use the Gauss-Newton updating direction. The constant information forgetting factor of Fortescue *et al.* (1981), the constant trace forgetting factor of Sripada *et al.* (1987) and a new forgetting factor based on maintaining a constant determinant are presented and compared by application to a simple simulated example using reduced rank data and to a closed loop experimental data set from a pilot scale distillation column. The effect of the choice of forgetting factor on the propagation of the P matrix is compared by examination of the parameter confidence bound ellipses.

Chapter 6 presents a new approach to the forgetting problem due to Kulhavy and Karny, (1984) which is substantially different from the exponential forgetting algorithms of Chapter 5. The motivating arguments for the algorithm are presented with an interpretation of the resulting equations. The algorithm is applied to the simulation and distillation column examples of Chapter 5 to provide a direct comparison.

Chapter 7 presents a review of various modifications to the basic least squares algorithm which result from the assumption that the process disturbance is bounded. The prediction error dead-zone method is discussed, and a set theoretic approach to identification is described. A modified least squares method of Dasgupta and Huang (1987) based on the bounded noise assumption which incorporates data-dependant updating and exponential forgetting is examined and reviewed with comparative examples.

Chapter 8 summarises the results of the entire thesis. Conclusions are drawn and recommendations for the direction of future work are outlined.

Chapter 2

Process Models For Self Tuning Control

2.1 Introduction

The field of process control relies on the use of linear, or linearised models of systems in order to describe dynamic behavior. The many available approaches to linear modeling, and their potential problems and pitfalls have led to many mis-applications and false notions in this field. For example the inappropriate treatment of non-zero mean steady states and low frequency disturbances by simple differencing of the data has been recognised as a problem in numerous works (Vermeer, 1987; Tuffs 1985).

This chapter focuses on the problems involved in the representation of discrete-time, time-invariant transfer function models to real-world processes. The broad objective is to provide a general overview of the models available and to present guidelines and recommendations for their use. The study is generally constrained to the choice of the "form", or structure of the model. The more specific problems related to modeling such as model order and delay determination or the choice of appropriate sampling rates are treated well elsewhere (eg. Ljung, 1987; Box and Jenkins, 1970), and hence are not discussed. The chapter covers the problem of including non-zero steady state levels in a dynamic model, and reviews numerous specific linear model structures from existing literature.

2.2 The Deterministic Dynamic Model

Linear, time-invariant systems form the most important class of dynamic systems considered for the design of controllers and in particular self-tuning controllers. A general description of a linear system is given as follows (Ljung, 1987):

> *A system is linear if its output response to a linear combination of inputs is the same linear combination of the output responses of the individual inputs. It is time-invariant if the output response to a certain signal does not depend on absolute time.*

Most real plant systems do not in general hold to these conditions, however a common and often valid approximation is to assume that they are linear in local regions of interest.

It is well known that a sampled linear time-invariant system whose input is constant between sample intervals may be described by its Markov sequence:

$$y(t)=G(q^{-1})u(t) \; ; \quad G(q^{-1})=\sum_{k=0}^{\infty} q^{-k} g_k \tag{2.1}$$

where y(t) and u(t) are the outputs and inputs of the model at discrete intervals of time denoted by t. This model is a rather inconvenient method of describing the process because it requires an infinite number of parameters. If $G(q^{-1})$ is a rational function then it may be factorised into finite numerator and denominator polynomials (Box and Jenkins, 1970). This more useful form of the linear model is the deterministic auto regressive moving average (DARMA) model which may be written as:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) \tag{2.2}$$

with $A(q^{-1})$ and $B(q^{-1})$ as polynomials in the backshift operator q^{-1}, given by:

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} \ldots a_nq^{-n} \tag{2.3a}$$
$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \ldots\ b_mq^{-m} \tag{2.3b}$$

The term d is the process delay specified as an integer number of sample intervals of t ($d \geq 1$). Equivalently, d may be interpreted as the number of leading terms of $G(q^{-1})$ that are equal to zero.

2.3 Disturbances

According to the deterministic model (2.1) the output of a system can be exactly calculated once the input is known. In practice however, there are usually unmeasurable signals such as noise and disturbances that affect the system. These may be accommodated by lumping them into a general disturbance term v(t):

$$y(t) = G(q^{-1})u(t) + v(t) \tag{2.4}$$

The lumped term v(t) considers the total effect of all classes of disturbances including load disturbances, measurement errors, parameter variations and noise.

The need for considering these disturbances in the model arises from the requirement to predict the future outputs based on information available at the present. When a model is to be used for control or prediction the effect of the unmeasurable signals on the prediction error (the difference between the actual future output and the predicted future output) must be canceled or eliminated for optimum performance (Åström and Wittenmark 1973). The model must therefore be capable of predicting the output based on some assumed knowledge of the disturbance behavior or based on some projection from a model of the past estimates of the disturbance. Since v(t) is assumed to be unmeasurable and of a general nature, a reasonable approach to its characterisation is to consider it as a random process (a sequence of random variables) and model and predict its behavior as the output of a linear filter driven by white noise. This idea was originally explored by Wold (1938) and has been much used in the study of linear dynamic systems, such as time series analysis (Box and Jenkins, 1970). A linear model of v(t) driven by a zero mean discrete white noise sequence, denoted by $\xi(t)$ is thus written as:

$$v(t) = H(q^{-1})\xi(t) \tag{2.5}$$

with $H(q^{-1})$ defined as:

$$H(q^{-1}) = 1 + \sum_{k=1}^{\infty} q^{-k} h_k \tag{2.6}$$

This yields a general linear stochastic model as:

$$y(t) = G(q^{-1})u(t) + H(q^{-1})\xi(t) \tag{2.7}$$

If $H(q^{-1})$ is considered to be a rational function then (2.5) may be factorised into finite numerator and denominator polynomials:

$$v(t) = H(q^{-1})\xi(t) = \frac{C(q^{-1})}{D(q^{-1})}\xi(t) \tag{2.8}$$

Such a form is known as the time series or auto-regressive moving average (ARMA) model (Box and Jenkins, 1970). The time series representation of general disturbances by filtered white noise is supported by the theorems of spectral factorisation which are developed from statistical and signal processing literature (Box and Jenkins, 1970). A basic theorem of spectral factorisation may be stated as the following (Åström and Wittenmark, 1984):

> *"Given a spectral density $\Phi(\omega)$ which is rational in $\cos(\omega)$, there exists a linear system with the pulse transfer function*
>
> $$H(z) = \frac{C(z)}{D(z)} \tag{2.9}$$
>
> *such that the output obtained when the system is driven by white noise is a stationary random process with spectral density Φ. The polynomial D has all its zeros outside the unit disk. The polynomial C has all its zeros outside the unit disk or on the unit circle."*

The proof and discussion of this theorem can be found in the literature on stochastic processes or stochastic control (Åström, 1970). It suffices to note that this theorem requires that the filter $H(q^{-1})$ be stable and its inverse be stable or marginally stable.

Note that choosing the leading term of $H(q^{-1})$ to be 1 does not reduce the generality of the description since the variance of $\xi(t)$ may take any value.

The invertability condition given by the theorem of spectral factorisation may be viewed as a condition on $H(q^{-1})$ for it to be useful as a predictor. For example suppose that the values of v(k) for $k \le t-1$ have been observed and it is required to predict the value of v(t). Such a prediction will be given by:

$$v(t) = \frac{C(q^{-1})}{D(q^{-1})}\xi(t) = \left[1-D(q^{-1})\right]v(t)+C(q^{-1})\xi(t) =$$

$$\xi(t)+c_1\xi(t-1)...+c_{nc}\xi(t-nc)-d_1v(t-1)...-d_{nd}v(t-nd) \tag{2.10}$$

The prediction (2.10) requires not only the value of $\bar{\xi}(t)$ (in this case $\xi(t)$ must be replaced by its expected value), but a means of determining the previous values of $\xi(k)$. Since $\xi(k)$ is not directly observable it must be estimated by inverting the noise model:

$$\xi(k) = \frac{D(q^{-1})}{C(q^{-1})}v(k);\ \forall\ k,\ k \le t \tag{2.11}$$

If $C(q^{-1})$ has all its roots inside the unit circle of the z-plane then the effect of the initial condition $\xi(0)$ on $\xi(k)$ will decay exponentially with time and may be ignored. However in the case that $C(q^{-1})$ has a root on (or near) the unit circle, equation (2.11) will be marginally stable and the influence of initial conditions $\xi(0)$ on $\xi(k)$ will not diminish (or diminish very slowly) with time. For example if $D(q^{-1})=1$ and $C(q^{-1})=1-q^{-1}$ then (2.11) may be written as:

$$\xi(k) = v(k) + \xi(k-1) \tag{2.12}$$

Attempting to calculate $\xi(k-1)$ from v(k-1), v(k-2)...v(0) gives:

$$\xi(k) = \sum_{i=0}^{k-1} v(k-i) + \xi(0) \qquad (2.13)$$

The presence of the unknown term $\xi(0)$ as $k \rightarrow \infty$ shows the consequences of C being unstable. Methods of handling this problem using a time-varying predictor such as a Kalman Filter have been presented (Åström and Wittenmark, 1984), however these authors point out that there are usually ways of reformulating the model to avoid using an unstable C. An example of such a situation is described in section 2.12.

The stability requirement on $H(q^{-1})$ given by the theorem of spectral factorisation implies that the disturbance will be bounded and:

$$1 + \sum_{k=1}^{\infty} |h_k| < \infty \qquad (2.14)$$

From (2.14) v(t) must be zero mean since:

$$E\{v(t)\} = H(1)E\{\xi(t)\} = \left[1 + \sum_{k=1}^{\infty} h_k\right] \cdot 0 = 0 \qquad (2.15)$$

The disturbance models for which this stability requirement is strictly held are termed stationary because for a stationary input signal such as white noise the output v(t) will also be stationary (Box and Jenkins, 1970). The term stationary in this context may be interpreted as meaning that the statistical properties given by the moments of the distribution function are constant for any arbitrary period of time over which the signal is examined. A signal is termed "weakly" stationary if only the first two moments (the mean and the variance) are constant.

2.4 Steady State Information and Low Frequency Disturbances

The dynamic model written in the form of (2.2) requires that the inputs and outputs have a constant ratio at steady state, given by the gain of the process:

$$\frac{y_{ss}}{u_{ss}} = \frac{B(1)}{A(1)} = \text{GAIN} \tag{2.16}$$

This is clearly not a reasonable constraint for the model of a process, because the input and output data in their raw or positional form, given by $u_m(t)$ and $y_m(t)$ are non-zero mean measurements of real quantities such as temperature or flow, whose steady state values do not generally have a constant "ratio" relationship as in (2.16).

Consider a linear process where the full scale deviation of 0 to 100% in control valve signal u_m causes a deviation of 80°C to 180°C in the measured temperature variable y_m. The gain of this process is unity because a change of one input unit gives a change of one output unit. Its steady state ratio given by (2.16) varies from infinity at u_{ss}=0% to 1.8 at u_{ss}=100%. A proportional controller acting on the measured signals $y_m(t)$ and $u_m(t)$ of this process would therefore have to be non-linear to operate properly, although this is clearly unnecessary because the dynamics are linear. The problem with using positional data is due to the offset or displacement of 80°C which exists on the measured variable.

Two different approaches may be applied to this problem:

1. Remove the steady state or nominal levels by pre-treatment of the measured data.

2. Consider the disturbance v(t) to have a non-zero mean component by modifying the structure of the disturbance model. This allows the model to be written in terms of positional data.

Several methods falling into each of these categories have been presented and evaluated for the case of off-line identification (Ljung, 1987). The first is perhaps more meaningful in terms of a physical interpretation and more analogous to the Laplace domain models which require that the input and output are exactly zero mean. The second, however must completely describe the steady states and load disturbances within its structure, hence it is useful in the design of regulatory controllers. A summary and discussion of these approaches is outlined in sections 2.5 and 2.6

2.5 Methods of Pre-Treating the Data

Removing the low frequency information by pre-treatment of the data is perhaps the most natural approach, since the objective is to build a dynamic model which is linearised about a particular steady state. In practice however it becomes difficult to draw a distinction between valuable low frequency information and the steady state or nominal levels. As a result the methods which fall into this category are generally *ad-hoc* in nature and often defy theoretical analysis. In general this approach differs from the second because the measured data $y_m(t)$ and $u_m(t)$ may not be directly applied to the model as the inputs and outputs y(t) and u(t). Several examples of data pre-treatment are described in the following sub-sections.

2.5.1 Deviations from Physical Equilibrium

The most natural approach is to determine an estimate of y_{ss} denoted by $\underline{y}$ which corresponds to a steady state value of u_m denoted by $\underline{u}$ close to the desired operating point. The values of $\underline{y}$ and $\underline{u}$ may be determined by a steady state model, or some historical knowledge of the operating conditions. Then define:

$$y(t) = y_m(t)-\underline{y}(t) \tag{2.17a}$$

$$u(t) = u_m(t)-\underline{u}(t) \tag{2.17b}$$

This approach, although perhaps the most desirable in terms of the physical interpretation, is impractical if steady state data are not available or if the steady states are affected by unmeasurable load changes.

2.5.2 The Incremental Approach

The low frequency information such as the steady state levels in a dynamic model may be eliminated by redefining y(t) and u(t) as the increments of the measured input and output:

$$y(t)= y_m(t)-y_m(t-1) = \Delta y_m \tag{2.18a}$$

$$u(t)= u_m(t)-u_m(t-1)= \Delta u_m \tag{2.18b}$$

Using the increments of the data Δu_m and Δy_m approximates the derivative of the input and output of the process whose values are zero at steady state. Such an approach is in practice identical to using a noise model with integration (Section 2.6b).

2.5.3 A Mean Deviational Approach

Another approach is to use the sample means as estimates of the steady states. The sample means are calculated from off-line or batch data as:

$$\underline{y} = \frac{1}{N}\sum_{k=1}^{N} y_m(k), \quad \underline{u} = \frac{1}{N}\sum_{k=1}^{N} u_m(k) \tag{2.19}$$

or in a recursive manner by the moving averages given by:

$$\underline{y}(t) = \alpha\underline{y}(t-1) + (1-\alpha)y_m(t-1); \quad 0 < \alpha \le 1 \tag{2.20a}$$

$$\underline{u}(t) = \alpha\underline{u}(t-1) + (1-\alpha)u_m(t-1) \tag{2.20b}$$

The model inputs and outputs y(t) and u(t) are defined as in (2.17). Such an approach is known as mean deviational. A problem with this method is that the estimated mean may not be a good estimate of the steady state. For example in the offline case the sample mean is constant while the steady states may be changing due to disturbances or setpoint changes within the data set.

In the recursive approach the moving average mean uses the parameter α to estimate the steady states, thus its behavior may vary depending on how α is chosen. Note that as α approaches zero the method becomes equivalent to using incremental data as in (2.18). Using the moving average mean equations (2.20) to estimate the steady states is also similar in practice to using a noise model with integration (Section 2.3.2b).

An ad-hoc approach similar to the above method was suggested (Tham *et al.* 1987) in which $\underline{y}$ is taken as the setpoint of a closed loop control system and $\underline{u}$ as calculated in (2.20b). This approach is based on the assumption that the steady state is indeed given by the setpoint (implying no control offset), but has the problem that the filter (2.20b) will introduce new dynamic modes to the system, since the mean deviational filter does not cancel out of the resulting closed loop transfer function.

2.6 Including the Displacement In the Disturbance Model

Considering the displacement to be a non-zero mean component of v(t) violates condition (2.14) and implies that $H(q^{-1})$ will not be a stationary process. This situation is usually remedied by either including an arbitrary constant value or closure term as part of v(t), or factoring one or more marginally stable components from $H(q^{-1})$ and modeling the resulting form of the disturbance as a stable process. The disturbance model approach differs from the methods of 2.5 in that it allows the measured variables $y_m(t)$ and $u_m(t)$ to be directly applied to the model inputs and outputs y(t) and u(t). The main advantage of this approach is that it models the steady states and load disturbances within the overall model structure, hence it is useful for the design of regulatory controllers by the "Internal Model Principle" (Francis and Wonham, 1976).

The following sections present several methods of including the steady state information in the model structure.

2.6.1 The Positional Model

A common simplification that may be made for linear processes that are not affected by load changes is to consider the steady states u_{ss} and y_{ss} to be linearly related, and combine them in a closure term μ. This term eliminates the need to consider the data in terms of deviations from a particular steady state. It may be interpreted as specifying a family of "constant load" steady states, and thus has the advantage of being unaffected by servo or setpoint changes. For the DARMA model (2.2) the value of μ is given by:

$$\mu = A(1)y_{ss} - B(1)u_{ss} \tag{2.21}$$

Rewriting equation (2.2) in terms of the positional data and including the μ term yields the "positional" form of the model:

$$A(q^{-1})y_m(t) = q^{-d}B(q^{-1})u_m(t) + \mu \tag{2.22}$$

In offline identification applications, this method of incorporating the steady states in the dynamic model is only a slight variant of the mean deviational methods described in section 2.5c (Ljung, 1987), hence the latter method is generally preferred because it requires fewer model parameters to be estimated.

2.6.2 A Noise Model with Integration

In contrast to the stability condition (2.14) stated for $H(q^{-1})$ it is often convenient to lump the nominal values y_{ss} and u_{ss} as non-zero mean components of v(t) by considering the noise model to be unstable.

The constant μ given by (2.22) may be modeled as a step disturbance of magnitude μ at time zero:

$$\frac{\mu}{1-q^{-1}}\,\delta(t) \tag{2.23}$$

where $\delta(t)$ is a unit impulse at time zero. If the value of μ is subject to random step changes at each interval then the closure term μ may be modeled as a series of step disturbances of random heights given by the sequence $\xi_1(t)$:

$$\mu(t) = \frac{1}{1-q^{-1}}\,\xi_1(t) \tag{2.24}$$

The disturbance v(t) may therefore be modeled as the sum of $\mu(t)$ and other strictly stationary disturbances given by $L(q^{-1})\xi_2(t)$ to give a marginally stable integrated disturbance model:

$$v(t) = \frac{1}{1-q^{-1}}\,\xi_1(t) + L(q^{-1})\xi_2(t) = \frac{C(q^{-1})}{\Delta}\xi(t) \tag{2.25}$$

The integrated disturbance model may be equivalently applied by forcing Δ to appear as a common factor of $A(q^{-1})$ and $B(q^{-1})$ by over parameterising each by an order of one. This is equivalent to considering Δ to be a factor of $D(q^{-1})$, with the disadvantage that it increases the number of parameters which must be estimated.

Using a disturbance model with integration such as (2.25) differs from the incremental data approach of 2.5b in that it allows the measured data to be directly applied to the model. A main advantage of this structure is that it allows the design of controllers with inherent reset or integral action (Clarke *et al.* 1983). This type of model is an important case which has been widely used for theoretical study and practical application (Vermeer, 1987). A more comprehensive discussion of its properties and applicability is the subject of Chapter 4.

A similar approach to the integrating noise model is the k-incremental process model (Clarke *et al.* 1983) in which the differencing operator Δ is replaced by a more general integrating term $1-q^{-k}$ with $k \geq 1$. This approach was considered in order to facilitate k-step ahead predictions of the process output in the design of a predictive controller (Tuffs and Clarke, 1985). The advantages of such an approach however are, for most practical situations, outweighed by the increased sensitivity to noise and the existence of better approaches to the multi-step prediction problem such as Generalised Predictive Control (Clarke *et al.* 1987).

The disturbance model with integration could be further generalised, to give more flexibility in describing the behavior of the disturbance by taking the polynomial $D(q^{-1})$ to be of higher order. For example periodic variations in the value of μ may be accounted for by allowing $D(q^{-1})$ to assume the form of a pair of complex poles on the unit circle (Goodwin *et al.* 1986). This general approach is suited to specific applications in which the disturbances may be well described by higher order transfer functions, however for most situations this is not the case, and the added complexity is not justified.

2.7 A Family of Transfer Function Models

The rational disturbance model defined by equation (2.8) may be included as an unmeasurable disturbance in a linear process model in different arrangements to describe a large class of disturbance situations. The following sections describe several important classes of models based on equations (2.8) and (2.4) that are worthy of consideration.

2.8 The Equation Error Description

A common approach to modelling the disturbance is to assume that it acts as an additional or exogenous input to a DARMA type system such as (2.2). These models are classed as equation error forms because the disturbance appears as a direct error in the difference equation of the model. For example the DARMA model with an equation error type disturbance is written as a difference equation as follows:

$$\begin{aligned} y(t) = {} & -a_1 y(t-1) \;\ldots - \; a_{na} y(t-n_a) \\ & + b_0 u(t-d) + b_1 u(t-d-1) + \ldots b_{nb}(t-n_b) + e(t) \end{aligned} \tag{2.26}$$

Where e(t) describes the overall effect of the disturbance. In such a model the polynomial $A(q^{-1})$ is a common factor in the denominators of $G(q^{-1})$ and $H(q^{-1})$.

A signal flow diagram for this situation is as depicted in Figure 2.1.

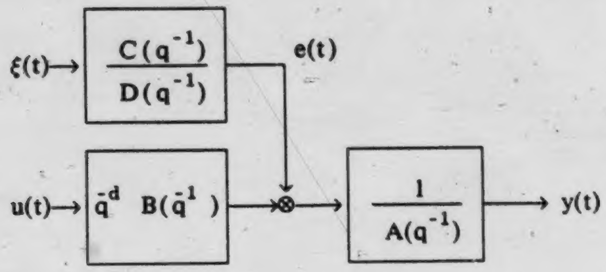

Figure 2.1 Equation Error Model Form

2.8.1 The ARX Form

The simplest form of an equation error model is that for which the disturbance is modeled as white noise:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t)+\xi(t) \tag{2.27}$$

Equation (2.25) is written in difference equation form as:

$$y(t) = -a_1 y(t-1)\ldots-a_{na} y(t-n)+b_0 u(t-d)\ldots+b_{nb} u(t-m-d)+\xi(t) \tag{2.28}$$

which corresponds to equation (2.7) with:

$$G(q^{-1}) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}\, u(t) \tag{2.29}$$

$$H(q^{-1}) = \frac{1}{A(q^{-1})}\, \xi(t) \tag{2.30}$$

This model has been widely studied in many contexts (Åström, 1968; Box and Jenkins, 1970) and is usually referred to as the ARX structure where AR refers to the auto-regressive part, $A(q^{-1})$, and X refers to the exogenous input $B(q^{-1})u(t)$. In the special case where the order of A (denoted by n_a) is zero, the model is known as the finite impulse response structure (FIR). This form has a large advantage over the ARX form in that it allows the identification regressor to be constructed of only delayed inputs, hence reducing the sensitivity of the parameter estimates to the effects of disturbances (see Section 3.7). Approximating an ARX process by a sufficiently high order FIR model eliminates the need for explicit knowledge of the order of the numerator and denominator polynomials, hence the FIR model is useful for describing unusual (high order) process dynamics (Ljung, 1987).

The ARX model form is probably the most favored for adaptive control applications, because its predictor defines a linear regression. The linear regression form allows the application of the method of least squares for the identification of the model parameters (this point is explained and further discussed in section 3.2). The applicability of least squares is a strong advantage for a model to possess, hence the ARX form is naturally appropriate for recursive identification applications, such as adaptive control.

2.8.2 The ARMAX Model Structure

The main disadvantage with the ARX structure is its lack of flexibility in describing the properties of the disturbance term. A simple method of adding flexibility to the disturbance is to consider $H(q^{-1})$ to be a moving average process (choose the order of D to be zero):

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t)+C(q^{-1})\xi(t) \tag{2.31}$$

This model is well known in statistical literature (Åström and Bohlin, 1965) as the "autoregressive moving average - exogenous variable" (ARMAX) model. It has been examined in terms of its relationship to an underlying continuous time (Laplace domain) process with the observation (Clarke, 1975; Åström, 1970) that the roots of $C(q^{-1})$ will always lie outside the unit circle of the z-plane.

2.8.3 Other Equation Error Forms

Other variations of the exogenous variable description may be derived by considering v(t) as a purely auto-regressive process, (Clarke, 1967; Cochrane and Orcutt, 1949) yielding the auto-regressive auto-regressive exogenous input (ARARX) structure:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \frac{1}{D(q^{-1})}\xi(t) \tag{2.32}$$

or considering v(t) as an auto-regressive moving average process yielding the ARARMAX (defined similar to ARMAX) structure (Ljung,1987):

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \frac{C(q^{-1})}{D(q^{-1})}\xi(t) \tag{2.33}$$

An important special case of (2.33) is that for which $D(q^{-1})$ is the differencing operator Δ. Such a model is termed the ARIMAX (auto-regressive integrated moving average with exogenous input) model or, in some control literature (Tuffs and Clarke, 1985) as the CARIMA (controlled-input auto-regressive integrated moving average) model:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \frac{C(q^{-1})}{\Delta}\xi(t) \tag{2.34}$$

The integrating noise model is further examined and interpreted in Chapter 4.

2.9 The Output Error Model Structure

The equation error model structures all correspond to descriptions where the transfer functions $G(q^{-1})$ and $H(q^{-1})$ have $A(q^{-1})$ as a common factor in their denominators. In many cases this restriction on the structure does not reflect the true disturbance mechanism, and a more realistic form would be to consider $G(q^{-1})$ and $H(q^{-1})$ as independent transfer functions. Such a structure is known as the output error model and the signal flow diagram for this configuration is given in Figure 2.2.

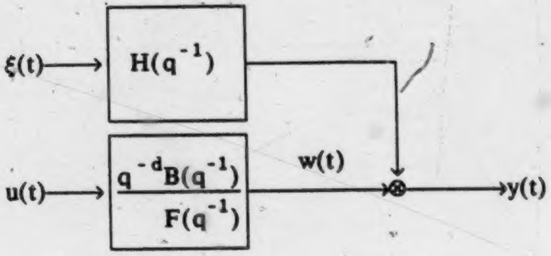

Figure 2.2 The Ouput Error Model Form

The main advantage of the output error form is that modelling errors of the plant or disturbance parts of the model do not affect (bias) the identification of the other part. Application of this structure effectively "decouples" the effect of the disturbance and input signals on the output (Ljung, 1987). For this reason the Box-Jenkins structure is a desirable structure for general use in off-line identification applications because the identification of the process polynomials ($B(q^{-1})$ and $F(q^{-1})$) will be "unbiased" by the disturbance even if the disturbance is not correctly modeled, or not modeled at all (eg. the OE model).

Appropriate application of the output error form can, in some situations, avoid introducing a noise model with numerator roots near the unit circle. For example consider a process whose ARMAX representation gives a $C(q^{-1})$ polynomial with roots near the unit circle:

$$(1 - 0.9q^{-1})y(t) = u(t-1) + (1 - 0.95q^{-1})\xi(t) \tag{2.35}$$

This process may be very closely approximated by the output error form:

$$y(t) = \frac{1}{1 - 0.9q^{-1}}u(t-1) + (1 - 0.05q^{-1})\xi(t) \qquad (2.36)$$

The disturbance for the output error model given by equation (2.36) is practically white noise and the roots of the numerator of the disturbance model are no longer near the unit circle. In general such situations result when the disturbance acts directly on the output measurements (eg. measurement noise).

2.9.1 The Measurement Noise Structure

The simplest form of output error form is the measurement noise case where it is assumed that the output y(t) is an undisturbed internal output w(t) that is corrupted by measurement error. In the simplest case white noise is assumed:

$$y(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t) + \xi(t) \qquad (2.37)$$

This model uses:

$$G(q^{-1}) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}; \quad H(q^{-1}) = 1 \qquad (2.38)$$

therefore it is a special model whose predictions do not depend on past values of the disturbance. The model given by equation (3.34) is often referred to as the output error (OE) structure (Ljung, 1987) and was used for the design of a model reference adaptive control scheme (Landau, 1976). Note that this model is identical to the ARMAX structure with $C(q^{-1})=A(q^{-1})$.

A general disadvantage of output error models is that the identification of their parameters is more difficult since w(t) is unobservable (see Sections 3.9 -3.10).

2.9.2 The Box-Jenkins Model Structure

A more general form of the output error model is developed by further modelling the properties of the output error. If we choose to describe it as an ARMA structure we obtain the model:

$$y(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}\xi(t) \qquad (2.39)$$

This is possibly the most natural finite dimensional parameterisation starting from (2.7) since $G(q^{-1})$ and $H(q^{-1})$ are independently parameterised as rational functions. It has been used (Bohlin, 1971) for general identification problems and (Young and Jakeman (1979)) for a refined instrumental variable recursive identification, however it was previously treated by, and is most commonly associated with the work of Box and Jenkins hence it is known as the Box-Jenkins structure.

2.10 The General Model Structure

Each of the previously discussed model structures may be considered as special cases of a general model description given by:

$$A(q^{-1})y(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}\xi(t) + \mu \tag{2.40}$$

where the value of μ in (2.40) is given by:

$$\mu = A(1)y_{ss} - \frac{B(1)}{F(1)}u_{ss} \tag{2.41}$$

if the positional form is required.

The model (2.40) is too general to be used in practicality. It is useful to consider the generalised case because algorithms and results generated from it also cover all the previously discussed cases which correspond to setting one or more of the five polynomials to unity or some known value (Ljung and Söderström, 1983). Some of the more commonly used special cases of (2.40) are summarised in Table 2.1.

Table 2.1 Some Common Special Cases of the General Model Structure

Special Case	Polynomials Used	Name of Model Structure
A=C=D=F=1	B	FIR Finite Impulse Response
C=D=F=1	AB	ARX Structure
D=F=1	ABC	ARMAX Structure
B=0 D=1	AC	ARMA Structure
C=F=1	ABD	ARARX Structure
F=1	ABCD	ARARMAX Structure
A=C=D=1	BF	Measurement Noise Structure
A=1	BFCD	Box-Jenkins
D=Δ F=1	ABC	ARIMAX or CARIMA structure
$\mu \neq 0$		Positional form

2.11 Summary

Chapter 2 presents a review and discussion of linear discrete models and provides design guidelines for the choice of suitable modelling approaches.

The DARMA model given by equation (2.2) is a convenient means of describing a deterministic system in a form which uses a finite (and usually small) number of parameters.

A common approach to modelling a stochastic process is to consider the unmeasured disturbance signal as a time series. The theorem of spectral factorisation allows such a sequence to be modeled as white noise, filtered by a rational transfer function, whose poles must be within the region of stability and whose zeros must be within or upon the stability boundary of the z plane. Disturbance models whose zeros are near the unit circle (stability boundary) should be avoided because in these cases the influence of the initial values of the noise sequence will linger for long periods of time. These cases are often indicative of an inappropriate choice of model structure.

A common problem in modelling real systems is that the input and output data have nominal, or steady-state values which are non-zero. Chapter 2 presents two main approaches to the problem:

1. Pre-treat the data by subtracting out estimates of the steady states, and re-defining the input and output of the model in terms of "deviations" from real or estimated physical equilibria.

2. Write the model in terms of the actual or "positional" data values by lumping the steady-state values in a displacement term μ. The displacement term may appear explicitly in the model or be considered part of a non-zero mean disturbance.

The former is more meaningful in terms of physical interpretation because it describes the data as "deviations" from some steady state level. This approach, however does not attempt to model low frequency disturbances such as load changes, and hence the controllers derived directly from the resulting models do not incorporate integral control.

The later however, requires that the model describes the load disturbances as well as the steady states by some internal model, hence the resulting model is useful in the design of regulators. For example a model which includes the differencing operator Δ as a factor of $D(q^{-1})$ will lead to the design of a controller with integral action to eliminate control offset due to load disturbances.

Model structures in wide use generally fall into one of two main classes based on different assumptions of how the disturbance affects the output. The equation error description takes the disturbance to be an extra input to the system, hence the process and disturbance models will have common poles (given by the $A(q^{-1})$ polynomial). This class is so named because the disturbance appears as an error term in the difference equation form of the model. The ARX model is perhaps the most important model structure of this class because its predictor defines a linear regression, allowing the method of least squares to be used for identification of the parameters. This structure is generally the most favored for adaptive control applications for this reason. The output error description is motivated by the assumption that the disturbance acts upon the measurement of the output by an independent mechanism, hence the process and disturbance parts of the model are independently parameterised. The output error (OE) and the Box-Jenkins (BJ) models are important examples in this class.

The choice of a model structure from one of the two broad classes should ideally be based on how well the physical situation conforms to the assumptions implied by that class, hence the choice of structure should be made based on an analysis of the sources of the disturbances affecting the output. Clearly, the situations whose disturbances affect the process as unmeasured inputs should be modeled using the equation error form, while situations where the disturbance affects the output by an independent process should be modeled using the output error form. In practice the choice is often made based on the identification tools available, however in these situations the designer is forced to make undesirable compromises if the algorithm is not sufficiently flexible or computing power is unavailable.

In general the output error form is preferred for off-line (or non-recursive) identification problems because the parameters of the process part of the model ($B(q^{-1})$ and $F(q^{-1})$) are unaffected by modelling errors in the disturbance part ($C(q^{-1})$ and $D(q^{-1})$) (Ljung, 1987). Application of an output error model to a system with input uncertainties may however require more parameters than the equation error model to describe the same system.

The output error structures however are less preferred for recursive applications because they require powerful iterative methods for high levels of accuracy and are more sensitive to errors in the model order and the delay (Ljung and Söderström, 1983). Furthermore, the corresponding recursive identification algorithms for the output error structures generally have slower convergence properties.

The equation error forms are generally preferred for recursive applications because the identification methods available for these are simpler to implement and have faster convergence properties. Applying the equation error model form to the independent measurement disturbance situation requires higher order disturbance models to be used, or will cause the process polynomials $A(q^{-1})$ and $B(q^{-1})$ to be biased, affects the measurement by an independent process.

A general model structure may be written which includes, as special cases, all of the models of the equation error and output error classes. The consideration of this structure allows a unified approach to the treatment of identification methods for both classes of model. Chapter 3 presents a review of prediction error identification methods using the general structure as a common basis for comparisons.

Chapter 3

Recursive Identification and Adaptive Filtering

3.1 Introduction

For many applications the model is needed to support decisions that must be made on-line, ie. during the operation of the system. For this reason it is often necessary to estimate the model parameters at the same time as the data is collected from the system. Such an approach to identification affords numerous advantages. The model parameters are ideally being continuously improved, and each measurement may be discarded after it has been applied to the identification algorithm. The need for storing a large batch of data is eliminated, allowing the algorithm to be applied to an indefinitely long series of data. A disadvantage of recursive methods is that the decision of model structure must be made a-priori, and in general, on-line comparisons of model structure are usually impractical.

The term "adaptive filtering" as applied to the recursive identification of a process model refers specifically to the identification of the disturbance model. Adaptive filtering is so named because the inverse of the disturbance model is applied as a filter to "adaptively" mask the effects of the disturbance on the data.

Adaptive control is a classic example of the use of recursive identification to provide a model to support on-line decisions pertaining to the control of the system. Recursive identification is considered central to the success or failure of adaptive control because the behavior of the estimation algorithm and hence the quality of the model will directly affect control performance.

It is useful to consider some definitions of the terminology and notation used in the following sections in order to facilitate the discussion and analysis of recursive identification. The system is defined as the mechanism from which the input and output signals are gathered. In the analysis of the properties of an algorithm it is often convenient to consider that the signals have been taken from a system which may be perfectly represented by a linear model with parameters denoted by θ_o. The parameters which "best" describe the system by a particular criterion are denoted by θ^*. This generally refers to the convergence point of a particular identification algorithm.

The choice of model structure for identification defines the parameter space $\mathbb{R}^n$ which is termed the model set, within which particular models may be chosen by giving values to the various parameters. Such a definition is useful because it allows any system to be classed as being within (or not within) the model set if the system may (or may not) be perfectly represented by one or more models in that set.

In the following sections a large family of recursive identification methods is considered within the framework of a general prediction error method approach. Important implementation aspects are considered, and recommendations for practical applications are provided.

3.2 The Prediction Error Methods in Identification

The linear regression methods applied to the models of Chapter 2 are based on the prediction of a dependant variable, $y(t)$ from a set of observations of the system at time t, given by a regressor or data vector $\phi(t)$:

$$\phi(t) = \left[\phi_1(t), \phi_2(t), \ldots \phi_n(t) \right]^T \qquad (3.1)$$

The linear regression method requires that such a prediction be written as a linear combination of the observations, or the inner product of a vector of parameters θ and the regressor $\phi(t)$:

$$\hat{y}(t|\theta) = \theta^T \phi(t) \qquad (3.2)$$

with

$$\theta = \left[\theta_1, \theta_2, \ldots \theta_n \right]^T \qquad (3.3)$$

(the argument $(t|\theta)$ implies that the value is a function of time conditioned on a particular value of θ)

The objective of a linear regression problem is to choose parameters θ to give estimates of y that are in some sense good. For instance one measure of the "goodness" or accuracy of fit is the prediction error given by:

$$\epsilon(t,\theta) = y(t) - \hat{y}(t|\theta) \qquad (3.4)$$

Prediction error methods are defined as linear regressions which choose $\hat{\theta}$ in order to minimise $V(\theta)$, the expected value of some scalar function of the prediction error f:

$$V(\theta) = E\left\{ f(t,\theta,\epsilon(t,\theta)) \right\} \qquad (3.5)$$

The parameter estimate $\hat{\theta}(t)$ is given by:

$$\hat{\theta}(t) = \arg\min\left\{V(\theta)\right\} \tag{3.6}$$

The function $f(\cdot)$ is a design parameter in prediction error identification methods. Certain choices of f lead to estimation algorithms with desirable properties such as an analytic solution of (3.6) or a solution which may be performed recursively. A reasonable approach to finding the best estimate $\hat{\theta}$ is to minimise the expected value of the square of the prediction errors, thus choose $f(\cdot)$ as:

$$f(t,\theta,\epsilon(t,\theta)) = \left[y(t)-\hat{y}(t|\theta)\right]^2 = \epsilon(t|\theta)^2 \tag{3.7}$$

This choice is the most common one made among a class of prediction error based methods of identification, although other variants have been suggested (Ljung and Söderström, 1983).

The prediction error methods based on the function (3.7) may be "fooled" by the existence of false local minima in the cost function $V(\theta)$. The following section provides a brief review of studies of the existence of local minima in order to provide guidelines for the choice of model structure.

3.3 False Local Minima

False local minima are local minima or valleys in the cost function that do not correspond to global minima. The existence of these is undesirable because they may prevent the identification algorithm from reaching the "best" parameters. There is no guarantee that a false local minimum will give a model that is good or even "acceptable". Some limited results are available concerning the existence of false local minima of the cost function $V(\theta)$ for various systems, subject to certain assumptions. These results provide useful guidelines for the choice of model structure. If the following conditions hold:

1. The inputs are persistently exciting

2. The system is operating in open loop, and may be perfectly described by:

$$A_0(q^{-1})y(t) = \frac{q^{-d}B_0(q^{-1})}{F_0(q^{-1})}u(t) + \frac{C_0(q^{-1})}{D_0(q^{-1})}\xi(t) \qquad (3.8)$$

3. The system is a member of the model set, ie. the system polynomials do not have higher degrees than the model polynomials.

then:

- If $B\equiv 0$ and $D\equiv F\equiv 1$ (ie. ARMA models) then there are no false local minima. (Åström and Söderström, 1974).

- If $C\equiv D\equiv F\equiv 1$ (the ARX model) then there are no false local minima since the cost function is quadratic in θ, and the minimum is unique.

- If C≡F≡1 (ARARX model) then there are no false local minima provided that the signal to noise ratio is sufficiently large. False minima may occur for noisy systems (Söderström, 1974)

- If A≡1 and n_f=1 (Output error or Box-Jenkins structures) then there are no false local minima. If n_f>1 then false local minima may exist in some cases (Söderström, 1975).

- If A≡C≡D≡1 (Output error model) and the input is white noise then there are no false local minima. For other inputs false local minima may exist (Söderström, 1975)

Recursive solutions to (3.6) which correspond to choosing the function $f(\cdot)$ as the square of the prediction error are special cases of prediction error methods which are examined in the following sections.

3.4 Recursive Prediction Error Methods

A recursive prediction error method (RPEM) is one that uses the prediction error multiplied by a gain vector, K(t) to provide the mapping of the regressor $\phi(t)$ to the parameter estimate $\hat{\theta}(t)$ at each interval. It has been shown (Ljung, 1978, Åström, 1980) that the general objective of such methods is the minimisation of a prediction error cost function ie. equation (3.6).

3.5 The Projection Algorithm

Using the regressor $\phi(t)$ multiplied by a positive constant k to obtain the gain K(t) was suggested as an algorithm for minimising the prediction error function (Robbins and Monro, 1951). This method is commonly known as the projection algorithm:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + k\phi(t)\epsilon(t) \tag{3.9}$$

The value of k may be chosen as simply a constant scalar or matrix, or a normalised gain factor:

$$k(t) = \frac{a}{b + \phi^T(t)\phi(t)} \tag{3.10}$$

(a and b are user chosen constants).

Algorithms of this form have been widely examined in the literature (eg. Goodwin and Sin, 1984; Shah, 1987) and have been found to possess interesting and desirable properties. The projection algorithm will converge under a very general set of conditions, and the algorithm gain does not diminish to zero (hence eliminating the need for forgetting factors). Convergence rates for these algorithms, however are relatively slow (Ljung and Söderström, 1983) thus they are usually only used in situations which require an algorithm of minimum complexity.

Another important sub-class of RPEMs is given by choosing the gain vector as the product of some non-singular positive definite matrix P and the negative gradient of the cost function with respect to the parameters, denoted by $-V'(\theta)$. If P is chosen to be an approximation of the inverse of the second derivative or "Hessian" matrix of $V(\theta)$ with respect to θ, then the numerical minimisation of $V(\theta)$ is similar to a "Newton Method" or "Quasi-Newton Method" (Ljung and Söderström, 1983). The Newton method may be summarised as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) - \frac{V'(t)}{V''(t)} \qquad (3.11)$$

For these methods the update of the parameters is very efficient in the neighborhood of a solution, but can be inefficient or even divergent when the parameters are not close to the solution. In practical algorithms the true Hessian matrix V" is approximated by a matrix with guaranteed positive definite conditions, thus assuring that the update is in the correct (downhill) direction. The following sections describe several important algorithms within this category.

3.6 The Method of Least Squares

Consider the problem of estimating the parameters of a single-input, single-output ARX model of the form:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + v(t) \qquad (3.12)$$

and assume that the variables u(t) and y(t) have steady state or nominal vales equal to zero. The ARX model (3.12) may be written as a linear regression problem:

$$y(t) = \theta^T\phi(t) + v(t) \qquad (3.13)$$

by choosing $\phi(t)$ and θ as:

$$\theta = \left[a_1 \ a_2 \ldots a_{na} \ b_0 \ b_1 \ldots b_{nb} \right]^T \qquad (3.14)$$

$$\phi(t) = \Big[-y(t-1)\ -y(t-2)\ldots -y(t-n_a)$$

$$u(t-d)\ u(t-d-1)\ldots u(t-d-n_b)\Big]^T \tag{3.15}$$

With no knowledge of v(t), a logical prediction of y(t) based on $\phi(t)$ and the parameter estimate $\hat{\theta}$:

$$\hat{y}(t|\theta) = \hat{\theta}^T\phi(t) \tag{3.16}$$

Approximating the cost function (3.5) by replacing the expected value by the weighted sample mean gives $V(\hat{\theta})$ as:

$$V(\hat{\theta}|t) = \frac{1}{t}\sum_{k=1}^{t}\alpha_k\epsilon(k|\hat{\theta})^2 \tag{3.17}$$

$$= \frac{1}{t}\sum_{k=1}^{t}\alpha_k(y(t)-\hat{\theta}^T(t)\phi(t))^2$$

where α_k is a series of scalar weighting factors. For simplicity of notation (and where not otherwise specified) it is assumed that the weighting sequence α is normalized such that $\sum_{k=1}^{t}\alpha_k = 1$. In recursive applications α_k is often chosen to discount or forget old or obsolete observations (see Section 4.2).

The criterion (3.17) may be analytically minimised to give an estimation equation:

$$\hat{\theta}_t = \left[\sum_{k=1}^{t} \alpha_k \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{t} \alpha_k \phi(k)y(k) \tag{3.18}$$

which yields the least squares solution. Equation (3.18) may be re-arranged to a recursive form by defining the matrix to be inverted as $\overline{R}(t)$ hence:

$$\overline{R}(t) = \sum_{k=1}^{t} \alpha_k \phi(k)\phi^T(k) = \overline{R}(t-1) + \alpha_t \phi(t)\phi^T(t) \tag{3.19}$$

The parameters may be updated recursively by:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \alpha_t \overline{R}^{-1}(t)\phi(t)\left[y(t)-\hat{\theta}^T(t-1)\phi(t)\right] \tag{3.20}$$

This equation is not well suited for online applications since the matrix $\overline{R}(t)$ must be inverted at each interval. The practical application of recursive least squares uses the inverse of $\overline{R}(t)$ denoted by P(t), which is updated by applying the matrix inversion lemma (Plackett, 1950) giving:

$$P(t) = P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t-1)P(t-1)}{1/\alpha_t + \phi(t)P(t-1)\phi^T(t)} \tag{3.21}$$

Updating P(t) by (3.19) has the advantage that the matrix inversion at each interval in equation (3.20) is replaced by a scalar division. The parameter update equations are:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left[y(t)-\hat{\theta}^T(t-1)\phi(t)\right] \tag{3.22}$$

$$K(t) = \alpha_t P(t)\phi(t) = \frac{P(t-1)\phi(t)}{1/\alpha_t + \phi(t)P(t-1)\phi(t)} \tag{3.23}$$

In equation (3.22) K(t) is a vector known as the Kalman gain. It is the determination of this vector that is the most powerful result of the recursive least squares (RLS) method, since it may be interpreted as an optimal solution to the recursive linear regression problem (Goodwin and Sin, 1984).

3.7 Asymptotical Properties of the RLS Algorithm

It has been shown (Ljung and Söderström, 1983) that the RLS estimates will asymptotically be equal to the off-line least squares estimates if the initial value of P(0) is chosen sufficiently large, and the weighting sequence α_t is chosen equivalently. The two methods are therefore asymptotically equivalent, and results obtained for the off-line version are applicable to the recursive version if those conditions are held. The criterion from which the least squares method was derived ensures that the prediction errors are minimised but does not guarantee that the parameter estimates $\hat{\theta}$ converge to θ_0. If the data have been generated by a system given by:

$$y(t) = \theta_0^T\phi(t) + v(t) \tag{3.24}$$

then the parameter estimates obtained by least squares are given by inserting the right side of equation (3.24) into equation (3.18) in place of y(t), and letting t become large. This gives an expression for the asymptotic value of $\hat{\theta}$:

$$\hat{\theta}(t) = \left[\sum_{k=1}^{t}\alpha_k\phi(k)\phi^T(k)\right]^{-1}\left\{\sum_{k=1}^{t}\alpha_k\left[\phi(k)\phi^T(k)\theta_0 + \phi(k)v(k)\right]\right\}$$

$$= \theta_0 + \left[\frac{1}{t}\sum_{k=1}^{t}\alpha_k\phi(k)\phi^T(k)\right]^{-1}\left\{\frac{1}{t}\sum_{k=1}^{t}\alpha_k\phi(k)v(k)\right\} \qquad (3.25)$$

If the disturbance v(t) is very small or zero then $\hat{\theta}$ will converge to some value close to or equal to θ_0. If there is any correlation between $\phi(t)$ and v(t) then

$$\lim_{t\to\infty}\left\{\frac{1}{t}\sum_{k=1}^{t}\alpha_k\phi(k)v(k)\right\} \neq 0$$

and there will be a bias between θ_0 and $\hat{\theta}(t)$. It has been demonstrated (Ljung and Söderström, 1983) that this bias will be zero in either of the following cases:

1) If v(t) is a sequence of independent random variables with a mean of zero (white noise). In this case v(t) will not depend on events previous to time t hence $E\{v(t)\phi(t)\} = 0$.

2) If u(t) is independent of v(t) and $n_a = 0$. In this case $\phi(t)$ will not contain terms which are correlated with v(t) thus $E\{v(t)\phi(t) = 0$.

For many applications it is not necessary for the parameters to converge to θ_0. A common objective, particularly in the field of predictive control, is to find a model of the system that gives the best prediction of the output. If the disturbance is a truly independent sequence then the minimisation of (3.18) guarantees that the best prediction is obtained. The following example (Ljung and Söderström, 1983 pp.196) presents a case where v(t) is a correlated (non-independent) sequence and convergence to the true parameters gives a higher prediction error variance than the least squares estimate:

Suppose that the true system is given by:

$$y(t) + ay(t-1) = bu(t-1) + \xi(t) + c\xi(t-1) \tag{3.26}$$

where u(t) and $\xi(t)$ are independent sequences of independent random variables with zero mean and unit variance. If an ARX model is applied to the system:

$$y(t) + \hat{a}y(t-1) = \hat{b}u(t-1) + \xi(t) \tag{3.27}$$

then the regressor $\phi(t)$ will be constructed as:

$$\phi(t) = \left[\, -y(t-1) \;\; u(t-1) \,\right]^T \tag{3.28a}$$

and the corresponding parameter vector will be:

$$\hat{\theta} = \left[\, \hat{a} \;\; \hat{b} \,\right]^T \tag{3.28b}$$

The variance of y(t) will be given from (3.26) as:

$$r_0 = E\left\{y^2\right\} = \frac{b^2 + c^2 - 2ac + 1}{1 - a^2} \tag{3.29}$$

The true or best prediction of y(t) from (3.26) is:

$$\hat{y}(t|\theta_0) = -ay(t-1) + bu(t-1) + c\epsilon(t-1) \tag{3.30}$$

where $\epsilon(t-1)$ is an estimate of $\xi(t-1)$ calculated from previous values of y, u and ϵ as:

$$\epsilon(t-1) = y(t-1) - \left[-ay(t-1) + bu(t-1) + c\epsilon(t-2)\right] \tag{3.31}$$

The actual disturbance is correlated because it contains terms in $\xi(t)$ and $\xi(t-1)$. The model does not account for this correlation directly but considers the overall effect of the disturbance within the data given by $\phi(t)$. The value of $\xi(t-1)$ is independent of u(t-1) but is related to y(t-1) (from (3.29)) by:

$$E\left\{\xi(t-1)|y(t-1)\right\} = \frac{1}{r_0}\, y(t-1) \tag{3.32}$$

Substituting this expected value into (3.30) for $\epsilon(t-1)$ gives the best or minimum variance prediction $y^*(t)$ within the model set:

$$\hat{y}^*(t) = -\left(a - \frac{c}{r_0}\right)y(t-1) + bu(t-1) \tag{3.33}$$

The minimum variance values of $\hat{a}$ and $\hat{b}$ are denoted by a^* and b^* and are given from (3.33) as:

$$a^* = a - \frac{c}{r_0} \qquad b^* = b \tag{3.34}$$

The value of the cost function based on these estimates is:

$$V(a^*,b^*) = E\left\{\epsilon(t|a^*,b^*)^2\right\} = 1 + c^2 - \frac{c^2}{r_0} \tag{3.35}$$

If the algorithm converged to the true parameters a,b then the cost function would be given by:

$$V(a,b) = E\left\{\epsilon(t|a,b)^2\right\} = 1 + c^2 \tag{3.36}$$

which is clearly greater than the minimum value given by (3.35). The method of least squares always gives a minimum variance prediction error thus the estimates will be given by a^* and b^* and since these are not the true parameters the estimates are said to be biased. The example clearly demonstrates that biased estimates are beneficial for the prediction.

In the previous example, the best or "true" predictor of the output (3.33) would have yielded an even smaller prediction error variance equal to unity if the model included c and the parameters approached the system parameters a,b and c. The least squares method however, cannot be applied to the problem of estimating c because the prediction error is a function of the model parameters and therefore is not a directly observable quantity. To include $\epsilon(t,\theta)$ in the regressor would violate the definition of a linear regression since the predictor will be a non-linear expression:

$$\hat{y}(t|\hat{\theta}) = \hat{\theta}^T\phi(t,\hat{\theta}) \tag{3.37}$$

The following section demonstrates a Newton method form of the RPEM which has been derived using approximations which allow consideration of more complicated model structures, such as the ARMAX model.

3.8 The Generalised Recursive Maximum Likelihood Method

In the method of least squares the criterion for how well the model performs is an approximation of the expected value of the square of the prediction error (3.17). This function could be analytically minimised in the RLS case because it was quadratic in the parameters. For more complicated models such as the ARMAX model this function is non-linear, and cannot be minimised analytically. In off line identification this problem is solved by using numerical minimisation techniques with several iterative passes through the entire data set. Such an approach is usually not feasible for on-line applications and as a result some approximations must be made to derive a Newton Method RPEM for these situations.

The recursive maximum likelihood (RML) method was originally derived for the ARMAX structure based on a Taylor series expansion of the cost function $V_t(\theta)$ about the estimate of the parameters $\hat{\theta}(t-1)$ (Åström and Bohlin, 1965). From this expansion a simple "Newton method" procedure is used to find $\hat{\theta}(t)$ that will approximately minimise that cost function, in a recursive fashion.

The following derivation demonstrates the approximations which are made to implement this recursive minimisation. It follows the derivation of the RML method presented in Ljung and Söderström (1983) but is not specific to the ARMAX structure. The results are applicable to the general model structure (2.40), hence the resulting algorithm is referred to in this work as the generalised recursive maximum likelihood (GRML) algorithm.

The Taylor series expansion of $V(\theta)$ about $\hat{\theta}(t-1)$ is written as:

$$V_t(\theta) = V_t(\hat{\theta}(t-1)) + V'(\hat{\theta}(t-1))\left[\theta - \hat{\theta}(t-1)\right] + \frac{1}{2}\left[\theta - \hat{\theta}(t-1)\right]^T V''_t(\hat{\theta}(t-1))\left[\theta - \hat{\theta}(t-1)\right] + \chi\left[|\theta - \hat{\theta}(t-1)|\right] \tag{3.38}$$

$\chi(x)$ denotes a residual function with the property that $\chi(x)/|x| \to 0$ as $|x| \to 0$. Minimisation of this function with respect to θ gives:

$$\hat{\theta}(t) = \hat{\theta}(t-1) - \left[V''_t(\hat{\theta}(t-1))\right]^{-1} V'^T_t(\hat{\theta}(t-1)) + \chi\left[|\theta - \hat{\theta}(t-1)|\right] \tag{3.39}$$

The negative derivative of $\epsilon(t,\theta)$ with respect to θ is known as the prediction error gradient and is defined as:

$$\psi(t,\theta) \quad -\frac{d}{d\theta}\,\epsilon(t,\theta) \tag{3.40}$$

Since the gradient defined by (3.40) and the prediction error are always functions of the parameters the notation $\psi(t,\theta)$ and $\epsilon(t,\theta)$ may for simplicity be replaced by $\psi(t)$ $\epsilon(t)$ with the implicit assumption of parameter dependence. Thus in the following references to these variables the parameter dependence is dropped where no ambiguity may exist.

An expression for $V'_t(\theta)$ may be written in terms of the gradient $\psi(t)$ by taking the derivative of the cost function from equation (3.17) with respect to θ:

$$\left[V'_t(\theta)\right]^T = \frac{dV}{d\epsilon}\frac{d\epsilon}{d\theta} = -\sum_{k=1}^{t} \alpha_t \psi(k)\epsilon(k) = \left[V'_{t-1}(\theta)\right]^T - \alpha_t \psi(t)\epsilon(t) \qquad (3.41)$$

and differentiating again gives a recursive expansion for V''_t:

$$V''_t(\theta) = V''_{t-1}(\theta) + \alpha_t(\psi(t)\psi^T(t) + \epsilon''(t)\epsilon(t)) \qquad (3.42)$$

Where $\epsilon''(t,\theta)$ is the second derivative matrix of $\epsilon(t)$ with respect to θ.

In order to evaluate $\hat{\theta}(t)$ from (3.39) a number of assumptions must be made to allow certain approximations:

1. The estimate $\hat{\theta}(t)$ is to be found in a small neighborhood of $\hat{\theta}(t-1)$, therefore $\chi(|\theta - \hat{\theta}(t-1)|)$ may be neglected, and $V''_t(\hat{\theta}(t)) \cong V''_t(\hat{\theta}(t-1))$.

2. Assume that the estimate $\hat{\theta}(t-1)$ is optimal at time t-1, thus $V'_{t-1}(\hat{\theta}(t-1)) \cong 0$.

3. Assume that the parameter estimates are close to the true parameters, therefore the prediction error $\epsilon(t)$ is approximately white noise, independent of events previous to time t-1, and in particular $\epsilon''(t)$, therefore a reasonable approximation is to take $\epsilon''(t)\epsilon(t)$ to be zero.

Using these approximations it is possible to evaluate a recursive estimate for the second derivative matrix $V''_t(\theta)$, which is denoted by $\overline{R}(t)$:

$$\overline{R}(t) = \overline{R}(t-1) + \alpha_t \psi(t)\psi^T(t) \qquad (3.43)$$

The second approximation allows (3.41) to be written:

$$V'_t(\hat{\theta}(t-1)) = -\alpha_t \psi(t)\epsilon(t)) \tag{3.44}$$

The parameter update expression (3.39) can be expressed as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \alpha_t \bar{R}^{-1}\psi(t)\epsilon(t) \tag{3.45}$$

For the least squares case (ie. an ARX model structure), all the above approximations are exact and the resulting algorithm (3.43) to (3.45) corresponds exactly to least squares, with the gradient $\psi(t)$ from (3.40) exactly equal to the regressor $\phi(t)$. The RML method was however specifically derived for more complex structures such as the ARMAX and therefore may be applied to the general model description (2.40), with the appropriate choice of $\psi(t)$. The next section demonstrates how the gradient may be chosen for the general model structure.

3.9 Choice of the Gradient for the General Model Structure

The gradient $\psi(t)$ is defined as the negative slope of the prediction error with respect to each of the parameters (equation (3.40). This is equivalent to the slope of the prediction $\hat{y}(t|\theta)$:

$$\psi(t) = -\frac{d\epsilon(t)}{d\theta} = \frac{d\hat{y}(t|\theta)}{d\theta} \tag{3.46}$$

A gradient $\psi(t)$ can therefore be written for any model structure if the prediction is differentiable in θ. For the general model structure (2.40) a differentiable expression for the prediction may be derived as follows. Arranging to a more useful form :

$$\frac{A(q^{-1})D(q^{-1})}{C(q^{-1})}\,y(t) = \frac{q^{-d}B(q^{-1})D(q^{-1})}{F(q^{-1})C(q^{-1})}\,u(t) + \xi(t) + \frac{D(q^{-1})}{C(q^{-1})}\,\mu \tag{3.47}$$

allows the predictor to be written as:

$$\hat{y}(t|\theta) = \left[1 - \frac{A(q^{-1})D(q^{-1})}{C(q^{-1})}\right]y(t) +$$

$$\frac{q^{-d}B(q^{-1})D(q^{-1})}{F(q^{-1})C(q^{-1})}\,u(t) + \frac{D(q^{-1})}{C(q^{-1})}\,\mu \tag{3.48}$$

The polynomials $A(q^{-1})$, $D(q^{-1})$ and $C(q^{-1})$ are all monic (leading terms equal to unity), thus the right hand side of (3.48) contains values of y only up to time t-1. If the roots of $C(q^{-1})$ and $F(q^{-1})$ are inside the unit circle of the z plane then the predictor will be stable and the effect of initial conditions on the predictions will diminish in time.

Equation (3.48) is still not easily differentiated in θ therefore the unobserved variables $\epsilon(t)$, $w(t)$ and $v(t)$ must be introduced. Re-arranging equation (3.45) to give the prediction error $\epsilon(t)$:

$$\epsilon(t) = \frac{D(q^{-1})}{C(q^{-1})}\left[A(q^{-1})y(t) - \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t) - \mu\right] \tag{3.49}$$

The undisturbed output state $w(t)$ and the disturbance $v(t)$ are estimated as $\hat{w}(t|\theta)$ and $\hat{v}(t|\theta)$ (for simplicity the parameter dependence notation is dropped where no ambiguity may exist):

$$\hat{w}(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t) \tag{3.50}$$

$$\hat{v}(t) = A(q^{-1})y(t) - w(t) - \mu \tag{3.51}$$

Combining equations (3.49), (3.50) and (3.51) gives an expression for the prediction error as:

$$\epsilon(t) = \frac{D(q^{-1})}{C(q^{-1})}\hat{v}(t) \tag{3.52}$$

The prediction $\hat{y}(t|\theta)$ required by equation (3.46) may be written as:

$$\begin{aligned}\hat{y}(t|\theta) &= \left[1 - A(q^{-1})\right]y(t) + \hat{w}(t) + \hat{v}(t) + \mu \\ &= -a_1 y(t-1) \ldots - a_{na} y(t-n_a) - f_1 \hat{w}(t-1) \ldots - f_{nf} \hat{w}(t-n_f) \\ &\quad + b_0 u(t-d) \ldots + b_{nb} u(t-d-n_b) + c_1 \epsilon(t-1) \ldots + c_{nc} \epsilon(t-n_c) \\ &\quad - d_1 \hat{v}(t-1) \ldots - d_{nd} \hat{v}(t-n_d) + \mu\end{aligned} \tag{3.53}$$

This may be written as the product of a parameter vector θ and a data vector $\phi(t|\theta)$ by choosing:

$$\theta = \left[a_1, \ldots a_{na}, f_1 \ldots f_{nf}, b_0 \ldots b_{nb}, c_1 \ldots c_{nc}, d_1 \ldots d_{nd}, \mu\right]^T \tag{3.54}$$

and

$$\begin{aligned}\phi(t|\theta) = \Big[&-y(t-1) \ldots -y(t-n_a), -\hat{w}(t-1) \ldots -\hat{w}(t-n_f), \\ &u(t-d) \ldots u(t-d-n_b), \epsilon(t-1) \ldots \epsilon(t-n_c), \\ &-\hat{v}(t-1) \ldots \hat{v}(t-n_d), 1\Big]^T\end{aligned} \tag{3.55}$$

The gradient $\psi(t)$ is given by the partial derivatives of (3.56) with respect to each element of (3.54):

$$\psi(t|\theta) = \left[\frac{\partial\hat{y}}{\partial a_1}\cdots\frac{\partial\hat{y}}{\partial a_{na}},\frac{\partial\hat{y}}{\partial f_1}\cdots\frac{\partial\hat{y}}{\partial f_{nf}},\frac{\partial\hat{y}}{\partial b_0}\cdots\frac{\partial\hat{y}}{\partial b_{nb}},\frac{\partial\hat{y}}{\partial c_1}\cdots\frac{\partial\hat{y}}{\partial c_{nc}},\frac{\partial\hat{y}}{\partial d_1}\cdots\frac{\partial\hat{y}}{\partial d_{nd}},\frac{\partial\hat{y}}{\partial\mu}\right]^T \quad (3.57)$$

which gives:

$$\frac{\partial\hat{y}}{\partial a_i} = -\frac{D(q^{-1})}{C(q^{-1})}\,y(t-i) \quad (3.58a)$$

$$\frac{\partial\hat{y}}{\partial b_i} = \frac{D(q^{-1})}{C(q^{-1})F(q^{-1})}u(t-d-i) \quad (3.58b)$$

$$\frac{\partial\hat{y}}{\partial f_i} = -\frac{D(q^{-1})}{C(q^{-1})F(q^{-1})}\,\hat{w}(t-i) \quad (3.58c)$$

$$\frac{\partial\hat{y}}{\partial c_i} = \frac{1}{C(q^{-1})}\,\epsilon(t-i) \quad (3.58d)$$

$$\frac{\partial\hat{y}}{\partial d_i} = -\frac{1}{C(q^{-1})}\,\hat{v}(t-i) \quad (3.58e)$$

$$\frac{\partial\hat{y}}{\partial\mu} = 1 \quad (3.58f)$$

The GRML algorithm therefore proceeds as follows:

1) The values of $y(t)$ and $u(t-1)$ are obtained.

2) The unobserved variables $w(t)$, and $v(t)$ are estimated from (3.49), (3.50) and (3.51) using the current parameter estimates.

3) The regressor $\phi(t)$ is constructed as in (3.55), and the prediction error is evaluated as:

$$\epsilon(t) = y(t) - \hat{\vartheta}^T(t-1)\phi(t) \tag{3.59}$$

4) The gradient $\psi(t)$ is generated as in equation (3.57) by filtering the data values using the filters given by equation (3.58) constructed from the parameter estimates at time t-1.

5) The covariance matrix P is updated as:

$$P^{-1}(t) = P^{-1}(t-1) + \alpha_t\psi(t)\psi^T(t) \tag{3.60}$$

or using the matrix inversion lemma:

$$P(t) = P(t-1) - \frac{P(t)\psi(t)\psi^T(t)P(t)}{1/\alpha_t + \psi^T(t)P(t)\psi(t)} \tag{3.61}$$

6) The parameters are updated by:

$$\hat{\vartheta}(t) = \hat{\vartheta}(t-1) + \frac{P(t-1)\psi(t)\epsilon(t)}{1/\alpha_t + \psi^T(t)P(t-1)\psi(t)} \tag{3.62}$$

3.10 *A Posteriori* Variables

An improvement may be made to step 2 of the algorithm by noting that the regressor $\phi(t)$ and the gradient $\psi(t)$ require the values of w, ϵ, and v only up to time t-1. It is reasonable therefore to calculate these values *a* *posteriori*, using the more recent set of parameters:

$$\hat{w}(t-1|\hat{\theta}(t-1)) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}u(t-1) \tag{3.63a}$$

$$\hat{v}(t-1|\hat{\theta}(t-1)) = A(q^{-1})y(t-1) - \hat{w}(t-1|\hat{\theta}(t-1)) - \mu \tag{3.63b}$$

$$\hat{\epsilon}(t-1|\hat{\theta}(t-1)) = y(t) - \hat{y}(t-1|\hat{\theta}(t-1)) \tag{3.63c}$$

The *a posteriori* prediction error given by (3.63c) is known as the residual. Using the residual in place of the prediction error in the calculation of the gradient has been examined theoretically and in simulation and it has been shown to give faster parameter convergence and greater accuracy (Ljung and Söderström, 1983). Use of *a posteriori* values has the most significant effect on the parameters that affect the prediction error non-linearly, eg. the parameters of $C(q^{-1})$ (Ljung and Söderström, 1983). The residual requires only slightly more computation than the *a priori* prediction error, and the *a posteriori* values of $\hat{w}(t-1)$ and $\hat{v}(t-1)$ require no additional calculations to evaluate *a posteriori* (they are not required for any other calculations). This improvement to the algorithm is widely accepted as common practice.

3.11 The Pseudo-Linear Regression

The simplicity of the linear regression method has led to the development of methods which attempt to cast models that are not "true" linear regressions into the linear regression form in order to directly apply the linear regression methods. This is commonly done by including in the regressor $\phi(t)$ unobservable variables whose values are estimated from the data and the most recent parameter estimates. The combined procedure of estimating θ by least squares and re-constructing the unobserved elements of $\phi(t)$ from the data and the parameters is commonly referred to as a pseudo-linear regression. (Ljung and Söderström, 1983; Solo, 1979). The most common of these is the extended least squares (ELS) method (Panuska, 1968) which is applicable to the ARMAX structure. The algorithm is implemented by choosing the regressor as:

$$\phi(t)=\begin{bmatrix} -y(t-1)\ldots y(t-n_a)\ u(t-d)\ldots u(t-d-n_b) \\ \epsilon(t-1)\ldots\epsilon(t-n_c) \end{bmatrix}^T \tag{3.64}$$

and

$$\theta = \begin{bmatrix} a_1\ldots a_2\ b_0\ldots b_{nb}\ c_1\ldots c_{nc} \end{bmatrix}^T \tag{3.65}$$

Parameter updating is by the usual RLS equations given by (3.21)- (3.23). This algorithm differs from the RML algorithm only in the way that the regressor is calculated. Comparing (3.64) to (3.55) gives the relation between $\phi(t)$ and $\psi(t)$ in the ELS case as:

$$\psi(t) = \frac{1}{C(q^{-1})}\phi(t) \tag{3.66}$$

Therefore the approximation made by using $\phi(t)$ in place of $\psi(t)$ is in taking $C(q^{-1})\cong 1$ in equation (3.66).

Another important PLR method is the Model Reference Output Error Method (Landau, 1976) which uses the model:

$$y(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}\, u(t) + v(t) \tag{3.67}$$

Such a model considers the existence of the undisturbed output w(t) given by:

$$w(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}\, u(t) \tag{3.68}$$

The regressor $\phi(t)$ is constructed using the estimates of the output $\hat{w}(t|\hat{\theta}(t-1))$ which are calculated using only the input sequence u(t) and previous output estimates:

$$\phi(t) = \left[\, -\hat{w}(t-1),\ldots -\hat{w}(t-n_f)\ u(t-d),\ldots u(t-d-n_b) \right]^T \tag{3.69a}$$

In this case θ is defined as:

$$\theta = \left[\, f_1 \ldots f_{nf}\ b_0 \ldots b_{nb} \right]^T \tag{3.69b}$$

The parameter estimates for this algorithm are given (Landau, 1976) by the usual RLS equations (3.19)- (3.21). This scheme has the advantage that the parameter estimates should be less sensitive to the properties of v(t) since n_a=0 (see Section 3.8). It may be viewed as an approximate gradient method since the true gradient $\psi(t)$ is actually given in terms of the regressor $\phi(t)$ as:

$$\psi(t) = \frac{1}{F(q^{-1})}\, \phi(t) \tag{3.70}$$

The approximation made by using $\phi(t)$ in place of the gradient $\psi(t)$ given by (3.70) is in taking $F(q^{-1}) \cong 1$.

The PLR methods are often termed approximate gradient methods (Ljung and Söderström (1983)) because the implicit dependence of the regressor $\phi(t|\theta)$ on the parameters is ignored in the calculation of the gradient, the general approximation is:

$$\psi(t) = \frac{d}{d\theta}\hat{y}(t|\theta) \cong \phi(t) \qquad (3.71)$$

Comparing this to (3.55) demonstrates that the quality of this approximation will depend on how close the polynomials C,D and F are to unity. The interpretation of the PLR methods as "approximations" does not necessarily imply that they are inferior. In a comparison between the RML and PLR methods it is concluded (Ljung and Söderström, 1983) that there are differences in convergence properties, algorithm complexity, asymptotical accuracy, and transient behavior and that there are certain advantages to each algorithm. The following sections summarise the convergence and asymptotical properties that have been derived for the RML and PLR algorithms.

3.12 Convergence and Asymptotical Properties of the RML Method

The RML method has been shown to converge to a local minimum of the selected cost function under a very general set of conditions (Ljung and Söderström, 1983).

For both the RML and the PLR methods a condition for convergence to the true parameters θ_0 was that the polynomials $C_0(q^{-1})$ and $F_0(q^{-1})$ be strictly stable, ie:

$$|C_0(e^{i\omega}) - 1| < 1 \;\forall\, \omega \tag{3.72a}$$

$$|F_0(e^{i\omega}) - 1| < 1 \;\forall\, \omega \tag{3.72b}$$

and that the input be persistently exciting and that the θ_0 belongs to the model set. It has been shown that the RML method does not require the last condition in order to converge to a local minimum of the cost function (Ljung and Söderström, 1983). This means that the algorithm will pick the best (measured by the performance criterion) approximation of the system within the model set. This value will however depend on the particular input used. The recursive algorithm converges to that approximation that is best under the input signal used in the experiment. If the true system is a member of the model set then the "best approximation" is equal to the true system description, and the algorithm will converge to that description regardless of the input, as long as it is sufficiently rich (Ljung and Söderström 1983).

3.13 Transient Behavior of the RML Algorithm

In practical application of the RML method it is clear that condition (3.72) must be held not only for the true system polynomials, but also for the estimated polynomials at each interval, since an unstable $\hat{C}(q^{-1})$ or $\hat{F}(q^{-1})$ will cause the gradient to become unbounded (this may be seen by examining equation (3.66)). This condition therefore requires that a stability constraint be placed on $\hat{C}(q^{-1})$ and $\hat{F}(q^{-1})$ at each interval to prevent unboundedness in the gradient. The practical necessity of this constraint has been demonstrated (Ljung and Söderström, 1983) and a projection method for its implementation is presented as the subject of Section 4.5.

In general the transient behavior of RML algorithms is inferior to the PLR algorithms. By comparative studies (Ljung and Söderström, 1983) it was concluded that the RML algorithms have slower convergence than the PLR algorithms, although the asymptotic accuracy is better. It was postulated (Ljung and Söderström, 1983) that the convergence is slower because using poor estimates of $C(q^{-1})$ and $F(q^{-1})$ for filtering the regressor (equations (3.64)- (3.68)) is actually worse than not filtering at all. They recommend that in order to speed convergence the filtering step (3.66)- (3.70) be omitted from the algorithm during the initial transient period.

3.14 Convergence and Asymptotical Results of the PLR Method

The PLR methods have been shown to converge to the true system description under slightly more restrictive conditions than the RML methods. Convergence has been explicitly proven only with the assumption that the system description is a member of the model set (Ljung and Söderström, 1983).

The PLR method was analysed by making a comparison to an associated differential equation (Ljung *et al.* 1975). Stability results could be proven by choosing explicit forms of this differential equation, however only the ELS ($F(q^{-1})=D(q^{-1})=1$) and the output error ($A(q^{-1})=D(q^{-1})=C(q^{-1})=1$) forms were examined. For these cases it was found that strict positive realness of

$C_0(q^{-1})$ and $F_0(q^{-1})$ was a sufficient but not necessary condition for convergence to the true values of the parameters. It was concluded (Ljung *et al.* 1975) that the PLR method will not converge if:

$$\mathrm{Re}\left[C_0(z)\right] < 0 \text{ for } z \in \mathbb{D} \tag{3.73a}$$

or

$$\mathrm{Re}\left[F_0(z)\right] < 0 \text{ for } z \in \mathbb{D} \tag{3.73b}$$

when $\mathbb{D}$ is a subset of the unit disk within which the polynomial $A_0(z)$ also has a zero. A numerical example of such a system was given (Ljung *et al.* 1975) as an ARMA process described by:

$$\begin{aligned} &y(t) + 0.9y(t-1) + 0.95y(t-2) \\ &\qquad = e(t) + 1.5e(t-1) + 0.75e(t-2) \end{aligned} \tag{3.74}$$

The $A_0(z)$ polynomial is in this case:

$$A_0(z) = z^2 + 0.9z + 0.95 \tag{3.75}$$

The zeros of $A_0(z)$ are:

$$z_{1,2} = -0.45 \pm 0.855i \tag{3.76}$$

The corresponding values of the C_0 polynomial are:

$$C_0(z_{1,2}) = -0.0845 \pm 0.713i \tag{3.77}$$

The ELS method applied to this example does not converge according to (3.73), since the real parts of $C_0(z)$ are less than zero at the root locations of $A_0(z)$. It was predicted (Ljung, 1975) that the values of the C parameters would oscillate below the true values with an approximate amplitude of 0.5. The period would vary with the log of time and would be slower and slower. It was later demonstrated (Ljung, 1980) that a full oscillation starting from t=1000 will take 150,000 time intervals.

The convergence or non-convergence of this particular example has been a contentious point in the literature (Panuska,1980a,b; Ljung,1975; Ljung,1980). It has been counter-argued (Panuska,1980a) that the example (3.74) is not truly described by the associated differential equations due to the roundoff effects of digital computer implementation. Simulation runs (Panuska, 1980) demonstrate that the example system converges to constant parameter values, because the gain becomes smaller than the minimum finite word length of the digital computer. Comparative simulations were presented (Ljung and Söderström, 1983) with the parameters initialised at their true values. In this demonstration the parameters from the RML method retained the true values while those from the ELS method drifted away from the true values.

It is generally agreed (eg. Ljung, 1980) that the non-convergence problem is rare in practice and that using a PLR on such a system compares favorably to the alternative of using RLS and identifying biased parameters (Panuska, 1980).

It has been demonstrated, (Solo, 1979) that implementing the stability constraint (3.72) on the parameter estimates is not necessary for the PLR methods, since for these the stability problem takes care of itself. A formal proof of this point has been presented (Solo,1979) however, the idea may be heuristically understood by considering that the existence of the prediction (as $\epsilon(t)$ or $\hat{w}(t)$ in the regressor) creates a feedback loop in the identification algorithm that is inherently stable (Solo, 1979). This stability guarantees that the predictor will be stable and will not converge to the stability boundary.

3.15 Generalised Least Squares

The generalised least squares (GLS) method (Hastings-James and Sage, 1969) is a variation of the RLS and RML methods that is applicable to the ARARX structure given by equation (2.32):

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \frac{1}{D(q^{-1})}\xi(t) \qquad (2.32)$$

The algorithm considers that the model given by (2.32) may be written in a linear regression form with an approximately white noise disturbance $v(t)$ in terms of filtered inputs and outputs:

$$A(q^{-1})y_f(t) = q^{-d}B(q^{-1})u_f(t) + v(t) \qquad (3.78)$$

The filtered data $y_f(t)$ and $u_f(t)$ are computed from the actual inputs and outputs using the current estimate of $D(q^{-1})$ as a filter:

$$y_f(t) = \hat{D}(q^{-1})y(t) \ ; \ u_f(t) = \hat{D}(q^{-1})u_f(t) \qquad (3.79)$$

If the disturbance v(t) is truly uncorrelated or "white" then the RLS method applied to the model (3.78) will give unbiased estimates of $A(q^{-1})$ and $B(q^{-1})$. The RLS method is applied to the model (3.79) by defining:

$$\phi_{AB}(t) = \left[-y_f(t-1)\ldots-y_f(t-n_a)\ \ u_f(t-d)\ldots u_f(t-d-n_b)\right]^T \tag{3.80}$$

and

$$\theta_{AB} = \left[\, a_1..a_{na}\ \ b_0\ldots b_{nb} \right]^T \tag{3.81}$$

and applying the parameter update equations:

$$\hat{\theta}_{AB}(t) = \hat{\theta}_{AB}(t-1) + P_{AB}(t)\phi_{AB}(t)[y_f(t) - \hat{\theta}^T_{AB}(t-1)\phi_{AB}(t)] \tag{3.82}$$

and

$$P_{AB}(t) = P_{AB}(t-1) - \frac{P_{AB}(t-1)\phi_{AB}(t)\phi^T_{AB}(t)P_{AB}(t-1)}{\alpha_t + \phi^T_{AB}(t)P_{AB}(t-1)\phi_{AB}(t)} \tag{3.83}$$

The polynomial $D(q^{-1})$ required in the filter equations (3.79) is estimated by calculating the model error of equation (3.78) in terms of the unfiltered data:

$$\epsilon(t) = A(q^{-1})y(t) - q^{-d}B(q^{-1})u(t) \tag{3.84}$$

The sequence $\epsilon(t)$ is modeled as an AR process:

$$D(q^{-1})\epsilon(t) = \xi(t) \tag{3.85}$$

The parameters of $D(q^{-1})$ are estimated by applying RLS to the model given by (3.85) by defining:

$$\phi_D(t) = \left[-\epsilon(t-1)\ldots\epsilon(t-n_d)\right]^T \tag{3.86}$$

and

$$\theta_d = \left[d_1\ldots d_{nd}\right]^T \tag{3.87}$$

The estimate of $D(q^{-1})$ is given by the RLS equations:

$$\hat{\theta}_D(t) = \hat{\theta}_D(t-1) + P_D(t)\phi_D(t)[y_f(t) - \hat{\theta}_D^T(t-1)\phi_D(t)] \tag{3.88}$$

and

$$P_D(t) = P_D(t-1) - \frac{P_D(t-1)\phi_D(t)\phi_D^T(t)P_D(t-1)}{\alpha_t + \phi_D^T(t)P_D(t-1)\phi_D(t)} \tag{3.89}$$

The RGLS algorithm defined by equations (3.79)- (3.89) is equivalent to an RML algorithm written for C≡F≡1 in which the regressor is formed as:

$$\phi^T(t) = \left[\ \phi_{AB}^T(t)\ \phi_D^T(t)\right] \tag{3.90}$$

and the off-diagonal blocks of the matrix $\phi^T\phi$ formed by the cross terms

$\phi_{AB} \times \phi_D$ are forced to zero (Ljung and Söderström, 1983 pp.116-117). The RGLS algorithm does not use approximate gradients as do the PLR methods therefore it is generally classified as an RML type method (Ljung and Söderström, 1983, pp.116-117).

3.16 The Instrumental Variables Method

In the discussion of the least squares method it was determined that θ_0 was the solution to the minimisation of the prediction error criterion:

$$\theta_0 = \min \arg E\left\{1/2\ \epsilon^2(t,\theta)\right\} \text{ iff } E\left\{\phi(t)v(t)\right\} = 0,\ \theta_0 \in \theta_M \tag{3.91}$$

This implies that v(t) must be zero mean and uncorrelated with $\phi(t)$, which will be true in the cases discussed in Section 3.8.

The Instrumental variables (IV) approach chooses a vector $\zeta(t)$ that is known to be uncorrelated with v(t) to replace $\phi(t)$ in the least squares parameter estimation algorithm. Appropriate choice of this vector should cause the parameter estimates $\hat{\theta}$ to approach the true system representation θ_0, if θ_0 is a member of the model set.

In order to demonstrate the motivation of the IV algorithm in the same way that the RPE methods were demonstrated consider the instrumental prediction error $e(t|\theta)$ given by:

$$e(t|\theta) = y(t) - \theta^T\zeta(t) \tag{3.92}$$

A prediction error cost function may be written to include e(t) and $\epsilon(t)$ as:

$$V(\theta) = E\left\{e\epsilon\right\} = E\left\{[y(t)-\theta^T\zeta(t)][y(t)-\theta^T\phi(t)]\right\} \tag{3.93}$$

Minimisation of (3.93) with respect to θ gives:

$$0 = E\left\{\zeta(t)[y(t)-\theta^T\phi(t)] + \phi(t)[y(t)-\theta^T\zeta(t)]\right\} \tag{3.94}$$

It is desired to minimise the parameter bias by eliminating the terms which contribute to the correlation between v(t) and $\phi(t)$. Since $\zeta(t)$ is chosen to be uncorrelated with v(t) the first term of (3.94) will not be affected by the disturbance. The second term however contains the term $\phi(t)y(t)$ which will contribute to the parameter bias. If the IV algorithm is therefore derived based on the first term of (3.94) only:

$$0 = E\left\{\zeta(t)[y(t)-\theta^T\phi(t)]\right\} \tag{3.95}$$

then the parameters should be unaffected by the disturbance, therefore unbiased.

The ordinary IV algorithm is derived from this cost function in a manner similar to the RLS method as:

$$\hat{\theta}(t) = E\left[\zeta(t)\phi^T(t)\right]^{-1} E\left[\zeta(t)y(t)\right] \tag{3.96}$$

or in a recursive form:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\zeta(t)\epsilon(t)}{1/\alpha_t + \phi^T(t)P(t-1)\zeta(t)}$$

$$= \hat{\theta}(t-1) + \alpha_t P(t)\zeta(t)\epsilon(t) \tag{3.97}$$

$$P(t) = P(t-1) - \frac{P(t-1)\zeta(t)\phi(t)P(t-1)}{1/\alpha_t + \phi^T(t)P(t-1)\zeta(t)} \tag{3.98}$$

If the actual system to be identified is defined by:

$$y(t) = \theta_0^T\phi(t) + v(t) \tag{3.99}$$

for which v(t) is a zero-mean, not necessarily white noise sequence then the parameter estimate vector $\hat{\theta}(t)$ will asymptotically approach the actual parameter vector θ_0 if the following conditions are satisfied:

$$\theta_0 \subset \theta_m$$

$$E\left\{\zeta(t)v(t)\right\} = 0 \tag{3.100}$$

and

$$\lambda_1 I \leq E\left\{\zeta(t)\phi(t)^T\right\} \leq \lambda_2 I \; ; \quad \lambda_2 > \lambda_1 > 0 \tag{3.101}$$

The last condition ensures that the P matrix is positive definite and that the input sequence u(t) is sufficiently rich.

The unbiased parameter properties of the IV methods allow the designer to ignore the properties of the noise model entirely provided that the steady states have been suitably dealt with. This usually leads to a simple choice of model structure such as the ARX form given by (2.27) or the measurement noise output error form given by (2.37).

A number of variations of the basic IV algorithm have been explored including the symmetric IV algorithm in which the regressor $\phi(t)$ is replaced by the instrumental vector $\zeta(t)$ in both positions of the IV information matrix, thus giving the new algorithm:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\phi(t)\epsilon(t)}{1/\alpha_t + \zeta^T(t)P(t-1)\zeta(t)} \qquad (3.102)$$

$$P(t) = P(t-1) - \frac{P(t-1)\zeta(t)\zeta(t)P(t-1)}{1/\alpha_t + \zeta^T(t)P(t-1)\zeta(t)} \qquad (3.103)$$

It has been observed (Young, 1984) that this form of the IV method is somewhat less reliable and slower to converge in practice. By performing a convergence analysis for this algorithm (Ljung and Söderström, 1983 pp.464-468) it was concluded that the asymptotic properties of the symmetric version coincide with the non-symmetric version under some restrictive assumptions regarding the choice of the instrumental vector $\zeta(t)$.

3.17 Choice of the Instrumental Vector

Many ways of choosing the instrumental variables $\zeta(t)$ have been proposed in the literature. Some typical choices have been reviewed (Ljung and Söderström, 1983; Söderström and Stoica, 1981). The elements of $\zeta(t)$ are usually formed from delayed and possibly filtered values of the input and outputs.

A common choice (eg. Finigan and Rowe, 1974) is to choose $\zeta(t)$ to correspond with $\phi(t)$ by replacing the delayed outputs with estimated outputs. These output estimates given by $x(t)$ are calculated from the inputs by a constant filter which is chosen as an *a priori* estimate of the plant model:

$$\zeta(t) = \left[-x(t-1) \ldots x(t-n_a) \;\; u(t-d) \ldots u(t-d-n_b)\right] \tag{3.104}$$

with:

$$\bar{A}(q^{-1})x(t) = q^{-d}\bar{B}(q^{-1})u(t) \tag{3.105}$$

It has been demonstrated (Ljung and Söderström pp.242-248) that an optimal or ideal choice of $\bar{A}(q^{-1})$ $\bar{B}(q^{-1})$ is $A_0(q^{-1})$ and $B_0(q^{-1})$ and that if these values were known then the IV estimate would have the highest accuracy of any estimation method for a reasonably posed problem (a general output error model structure eg. Figure 2.2). Such an algorithm is of course impractical because the true system parameters are unknown to the designer. A practical variation of this idea however is to generate x(t) by adaptive filters; eg. the last available estimates are used to compute the instrumental variables. Thus $\zeta(t)$ would be given by (3.104) with x(t) generated by:

$$x(t) = \hat{\theta}^T(t-1)\zeta(t) \tag{3.106}$$

This choice has been proposed (Wong and Polack, 1967; Young, 1970) with variations to improve stability such as using delayed and filtered parameter estimates θ_{aux} in place of $\hat{\theta}(t-1)$ ie:

$$x(t) = \theta^T_{aux}(t-d)\zeta(t) \tag{3.107}$$

$$\theta_{aux}(t) = (1-\beta)\theta_{aux}(t-1) + \beta\hat{\theta}(t-1); \; \beta\in(0.02,\ 0.05) \tag{3.108}$$

This choice of the IV algorithm has been compared to other identification methods (Isermann et al ,1974) and found to be reasonably robust and accurate.

A refined IV (RIV) method has been proposed (Young, 1976) in which the data are prefiltered by a constant filter $T(q^{-1})$ which is an approximation of the disturbance model $H(q^{-1})$. It is implemented as follows:

$$y_f(t) = \frac{1}{T(q^{-1})}\, y(t); \quad \phi_f(t) = \frac{1}{T(q^{-1})}\, \phi(t) \tag{3.109}$$

$$\epsilon_f(t) = y_f(t) - \phi_f^T(t)\hat{\theta}(t-1) \tag{3.110}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\zeta(t)\epsilon_f(t)}{1/\alpha_t + \phi_f^T(t)P(t-1)\zeta(t)} \tag{3.111}$$

$$P(t) = P(t-1) - \frac{P(t-1)\zeta(t)\phi_f(t)P(t-1)}{1/\alpha_t + \phi_f^T(t)P(t-1)\zeta(t)} \tag{3.112}$$

An extension of the RIV algorithm is the RIV-AML (Refined instrumental variable approximate maximum likelihood) (Young, 1976) method in which the model is assumed to be a Box-Jenkins structure (equation (2.39)):

$$y(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}\, u(t) + \frac{C(q^{-1})}{D(q^{-1})}\, \xi(t)$$

An IV algorithm is used to estimate the dynamic part $q^{-d}B(q^{-1})u(t)/F(q^{-1})$ while the disturbance part $C(q^{-1})\xi(t)/D(q^{-1})$ is estimated by modeling the difference $y(t)-w(t)$ as an ARMA process using the ELS method (the term AML (approximate maximum likelihood) is a synonym for ELS). The Box-Jenkins Structure is re-arranged to the regression form as:

$$F(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + F(q^{-1})v(t) \tag{3.113a}$$

with

$$v(t) = \frac{C(q^{-1})}{D(q^{-1})}\,\xi(t) \tag{3.113b}$$

The filtered data are given by:

$$y_f(t) = \frac{D(q^{-1})}{F(q^{-1})C(q^{-1})}\,y(t); \tag{3.114a}$$

$$\phi_f(t) = \frac{D(q^{-1})}{F(q^{-1})C(q^{-1})}\,\phi(t) \tag{3.114b}$$

$$\zeta_f(t) = \frac{D(q^{-1})}{F(q^{-1})C(q^{-1})}\,\zeta(t) \tag{3.114c}$$

with:

$$\phi(t) = \left[-y(t-1)\ldots y(t-n_f)\ u(t-d)\ldots u(t-d-n_b)\right]^T \tag{3.115a}$$

$$\zeta(t) = \left[-w(t-1)\ldots w(t-n_f)\ u(t-d)\ldots u(t-d-n_b)\right]^T \tag{3.115b}$$

$$w(t) = \frac{q^{-d}B(q^{-1})}{F(q^{-1})}\,u(t) \tag{3.116}$$

The IV estimation of $B(q^{-1})$ and $F(q^{-1})$ is performed by defining:

$$\theta_{BF} = \left[f_1 ... f_{nf} \; b_0 ... b_{nb} \right]^T \tag{3.117}$$

and applying the RIV equations (3.110), (3.111) and (3.112) with $\zeta(t)$ replaced by $\zeta_f(t)$, and $\phi_f(t)$, $\epsilon_f(t)$ and θ defined by equations (3.114) and (3.117).

The disturbance model given by equation (3.113b) is estimated by calculating the disturbance v(t):

$$\hat{v}(t) = y(t) - w(t) \tag{3.118}$$

and applying the method of ELS to the estimation of the ARMA model given by:

$$v(t) = \frac{C(q^{-1})}{D(q^{-1})} \xi(t) \tag{3.119}$$

Such an estimation is carried out by defining:

$$\epsilon_v(t) = v(t) - \hat{\theta}^T{}_{CD}(t) \phi_{CD}(t) \tag{3.120}$$

$$\phi_{CD} = \left[-v(t-1) ... -v(t-n_d) \; \epsilon_v(t-1) ... \epsilon_v(t-n_c) \right]^T \tag{3.121a}$$

$$\theta_{CD} = \left[d_1 ... d_{nd} \; c_1 ... c_{nc} \right]^T \tag{3.121b}$$

and updating $\hat{\theta}_{CD}(t)$ by:

$$\hat{\vartheta}_{CD}(t) = \hat{\vartheta}_{CD}(t-1) + \alpha_t P(t)\phi(t)\left[v(t)-\hat{\vartheta}^T_{CD}(t-1)\phi_{CD}(t)\right] \tag{3.122}$$

$$P(t)= P(t-1) - \frac{P(t-1)\phi_{CD}(t)\phi^T_{CD}(t-1)P(t-1)}{1/\alpha_t + \phi_{CD}(t)P(t-1)\phi^T_{CD}(t)} \tag{3.123}$$

Of course the true values of the polynomials $F(q^{-1})$, $C(q^{-1})$ and $D(q^{-1})$ are unavailable for use in equations (3.114) and (3.116) thus they are replaced by the current estimates.

The RIV-AML algorithm has been shown to have greater statistical efficiency but is less robust than the comparable IV methods (Young, 1984 pp.175-204). It is recommended for situations where reasonable estimates of the $B(q^{-1})$ and $F(q^{-1})$ polynomials are available and greater statistical efficiency is required.

3.18 Summary

The many variations available for choice of model structure, described in Chapter 2 generate a need for "customised" approaches to the recursive parameter identification problem. Recursive prediction error methods have been widely studied in the literature, and discussed in Chapter 3. A comprehensive treatment of the subject has been developed in the form of numerous algorithms, each of which may be interpreted as a variation or extension of the basic recursive least squares method.

In practice it is intuitively favorable to a choose a system model whose corresponding adaptation algorithm is of minimal complexity, and identifies the minimum number of parameters. A practical algorithm must posses favorable convergence properties, such as an efficient rate of convergence, and a guarantee that the parameters will asymptotically converge to constant values. Indeed, it is often the case that the choice of model structure is based on or at least influenced by the availability of an identification algorithm with these properties. For instance, the output error model structures cannot be directly written in a linear regression form hence they are often avoided in favor of the equation error models.

RLS is a prediction error identification algorithm which updates the parameters using an estimate of the "Newton Method" direction. Since RLS is only applicable to the ARX structure, various approximations have been invoked in order to apply its principles to more complicated structures. These approximate methods are generally derived from off-line methods. Chapter 3 describes three main classes of these algorithms:

1) The GRML method defined by equations (3.55)- (3.63) is applicable to each of the model structures discussed in Chapter 2. It has been demonstrated (Ljung and Söderström, 1983) to possess valuable asymptotic properties under relatively general conditions, however convergence rates are generally slower than the PLR methods. GLS is a variant of the GRML applied to the ARARMAX model which constrains the off-diagonal blocks of the matrix $\phi^T\phi$ formed by the cross terms $\phi_{AB}\times\phi_D$ to zero.

2) The PLR method is an approximation of the GRML in which the gradient vector $\psi(t)$ is replaced with the data vector $\phi(t)$. ELS is a PLR method applied to the ARMAX structure. ELS has been demonstrated (Ljung and Söderström, 1983) to converge faster than the RML method, and have the additional advantage of inherent stability.

3) IV and RIV methods are a modification of least squares which are designed to give "unbiased" estimates of the parameters. They generally

employ an instrumental vector $\zeta(t)$ constructed from filtered or delayed inputs. IV methods are, in general, slower to converge and lacking in inherent stability in comparison to the PLR methods, however their properties of asymptotic accuracy make them valuable in many applications.

In general, the more complicated identification algorithms (eg. GRML, PLR, IV) should be avoided in favor of RLS whenever possible because for real-world problems the properties of the disturbance are often not uniform, and the noise models are difficult to estimate by recursive methods. In choosing the model to be identified it is always advisable to incorporate as much *a priori* knowledge of the system as possible in order to reduce the number of parameters to be identified. The integrating noise model is a special application of *a priori* knowledge to the disturbance model, ie. the assumption that the disturbance model has a pole on the unit circle. Chapter 4 presents a more detailed analysis of its properties and applicability.

Chapter 4

Application of the Differencing Operator

4.1 Introduction

The differencing operator Δ ($\Delta = 1-q^{-1}$) is commonly applied to the input and output measurements to eliminate the closure or displacement term μ from the postional form of the dynamic model. In off-line identification analysis (Box and Jenkins, 1970) differencing is recommended when the input or output series are non-stationary. In adaptive control work differencing has been introduced arbitrarily to simplify controller design (Tuffs *et al.*, 1985, Vermeer *et al.*, 1987). In this a common assumption made is that the C polynomial is equal to unity (Vermeer *et al.*, 1987). However in processes with significant levels of stationary disturbances such as measurement noise the differencing approach leads to poor results (Wahlberg and Ljung, 1986) because it increases the sensitivity of the estimator to high frequency components in the data. In the presence of stationary noise disturbances the C polynomial has an important role in "filtering" the effects of stationary noise on the non-stationary random walk disturbance process. The C polynomial allows the ARIMAX structure to model the combined effects of filtered "white noise" and a "Brownian motion" disturbance component in the data.

In this chapter it is demonstrated that independent sources of stationary and non-stationary noise will cause the C polynomial to have positive, real roots and the location of these roots represent the proportion of stationary to non-stationary noise.

This observation sheds some understanding on the C polynomial and allows the designer to make reasonable *a priori* choices or at least place reasonable boundaries on the parameters of C based on knowledge of the disturbance and noise characteristics of the system.

It is demonstrated that the true or "best" choice of the $C(q^{-1})$ polynomial for a particular system is equivalent to a Kalman filter for tracking the value of the time-varying closure term μ. Furthurmore, the ELS and RML schemes are methods for estimating the parameters of that Kalman filter.

4.2 The Integrating or Brownian Motion Noise Model

The Brownian motion noise model $\frac{1}{\Delta}\xi(t)$ is useful for eliminating the steady-state displacement term μ from the model because it describes the disturbance in terms of a series of random steps. In practical situations however this description is inadequate, because the measurements are typically affected by various independent disturbances each of which may be stationary or non-stationary. Some examples of those disturbances which fall into the stationary category are measurement noise or small fluctuations in process conditions about a nominal level, perhaps due to inadequate mixing. The latter type are usually seen as process load changes, such as feed disturbances or changes in ambient conditions which affect the process output.

Figure 4.1 illustrates a system representation with independent stationary and non-stationary disturbances.

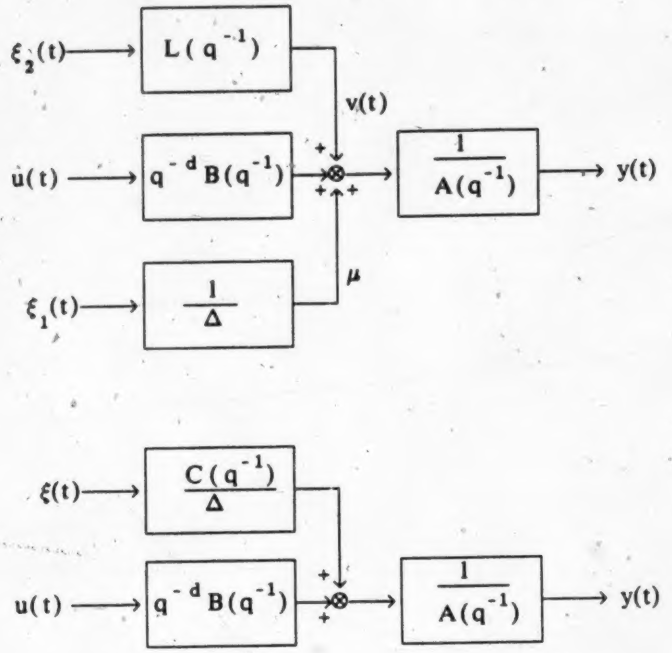
$\xi_2(t)$
$L(q^{-1})$
$v(t)$
$u(t)$
$q^{-d}B(q^{-1})$
+
+
+
$\frac{1}{A(q^{-1})}$
$y(t)$
$\xi_1(t)$
$\frac{1}{\Delta}$
μ
$\xi(t)$
$\frac{C(q^{-1})}{\Delta}$
$u(t)$
$q^{-d}B(q^{-1})$
+
+
$\frac{1}{A(q^{-1})}$
$y(t)$

Figure 4.1 Equivalent Representations of the ARIMAX Structure

If the disturbance sequences ξ_1 and ξ_2 are completely independent then they may be combined into the disturbance term $\frac{C(q^{-1})}{\Delta}\xi(t)$. With such a factorisation the ARIMAX model is written as:

$$A(q^{-1})\ y(t) = q^{-d}B(q^{-1})\ u(t) + L(q^{-1})\xi_2 + \frac{M(q^{-1})}{\Delta}\xi_1 \tag{4.1}$$

where $L(q^{-1})$ and $M(q^{-1})$ are stable, monic polynomials. In the simple case when $L(q^{-1}) \equiv M(q^{-1}) \equiv 1$, the polynomial $C(q^{-1})$ will be first order and the value of c_1 may be calculated in terms of the variances of ξ_1 and ξ_2 denoted by σ_1^2 and σ_2^2 from:

$$(1 + c_1 z)(1 + c_1 z^{-1})\sigma^2 = (1 - z)(1 - z^{-1})\sigma_2^2 + \sigma_1^2 \tag{4.2}$$

Equating terms in similar powers of z gives:

$$c_1 = -\frac{\sigma_2^2}{\sigma^2} \tag{4.3}$$

and:

$$(1 + c_1^2)\sigma^2 = 2\sigma_2^2 + \sigma_1^2 \tag{4.4}$$

Combining these gives a quadratic expression for c_1 in terms of σ_2^2 and σ_1^2 for which all the solutions are real and the root pairs c_1 and c_1' correspond to the roots of $C(z)$ and $C(z^{-1})$:

$$C(z)C(z^{-1}) = (1 + c_1 z)(1 + c_1 z^{-1}) \tag{4.5}$$

$$\sigma_2^2 c_1^2 + (2\sigma_2^2 + \sigma_1^2)c_1 + \sigma_2^2 = 0 \tag{4.6}$$

$$c_1, c_1' = -1 - \frac{\sigma_1^2}{2\sigma_2^2} + \frac{\sigma_1}{2\sigma_2^2}\sqrt{4\sigma_2^2 + \sigma_1^2} \tag{4.7}$$

The roots c_1 and c_1' are a symmetric pair ($c_1 \cdot c_1' = 1$). They are real and less than zero because σ_2^2 and σ_1^2 are positive. This constrains the value of c_1 to the range (0, -1) because only the stable solutions of c_1 are admissible. A negative value of c_1 corresponds to a positive root of the $C(q^{-1})$ polynomial.

In the first order case the value of c_1 may be interpreted as a measure of the relative magnitudes of the stationary and non-stationary noise components, with $c_1 = 0$ corresponding to "pure" Brownian motion and $c_1 = -1$ corresponding to white noise. Positive values of c_1 mean that the disturbance cannot be factorised into independent stationary and non-stationary modes. This observation can be extended to higher order C polynomials describing numerous independent inputs. Assuming that the non-stationary component of the disturbance is a single Brownian motion signal, then all the roots of the C polynomial will be positive, and the number of roots will be given by the number of independent inputs. This is demonstrated by the following example. Consider a system with three independent inputs, two white signals and one Brownian motion signal:

$$v(t) = \xi_1 + \xi_2 + \frac{\xi_3}{\Delta}$$

The stationary sequences ξ_1 and ξ_2 may be equivalently described by a single filtered signal:

$$v(t) = (1 + r_1 q^{-1})\xi_4 + \frac{\xi_3}{\Delta}$$

or partially combined into a filtered signal $(1 + r_1 q^{-1})\xi_5$ and a white signal ξ_6. This partial combination allows the value of r_1 to be chosen to allow the following spectral combinations:

$$v(t) = (1 + r_1 q^{-1})\xi_5 + \left[\xi_6 + \frac{\xi_3}{\Delta}\right]$$

$$v(t) = (1 + r_1 q^{-1})\xi_5 + \frac{(1 + r_1 q^{-1})\xi_7}{\Delta} = (1 + r_1 q^{-1})\left[\xi_5 + \frac{\xi_7}{\Delta}\right]$$

$$= \frac{(1 + r_1 q^{-1})(1 + r_2 q^{-1})}{\Delta}\xi_8$$

This example demonstrates how the order and the root locations r_1 and r_2 of are related to a description of a system with multiple independent inputs.

4.3 Frequency Response Properties of the Differencing Operator

A considerable deterrent to the use of the differencing operator in process identification is that it places undue emphasis on the frequency bandwidth close to the Nyquist frequency. This is a problem in process control applications for two reasons:

1) Typically the model is intended as low order approximation of a system which has high frequency dynamics and non-linearities. In these cases it is important that the model is fitted to the lower frequency bandwidth (Ljung, 1987).

2) High frequency bandwidths generally contain a higher proportion of noise and random effects hence the models are less accurate.

To understand the effect of the differencing operator as a high-pass filter consider its gain as a function of frequency given by:

$$\mathrm{Gain}(G(q^{-1})) = |G(e^{-i\omega T_s})|$$

$$\mathrm{Gain}(\Delta) = |(1-e^{-i\omega T_s})|$$

$$= |1-\cos(\omega T_s)+i\sin(\omega T_s)| = \sqrt{(1-\cos(\omega T_s))^2 - \sin^2(\omega T_s)} \qquad (4.8)$$

This function is plotted in Figure 4.2a. For frequencies approaching zero, the gain of the differencing filter tends to zero, effectively removing the steady state information. The gain increases however, with frequency to a maximum of 2 occurring at the Nyquist frequency, $\omega T_s = \pi$. The result is that increased emphasis or weighting is placed on the high frequency components when the data is differenced. This amplification effect is undesirable because high frequency bandwidths (approaching the Nyquist frequency) often contain significant proportions of noise, and these are usually not of interest for control purposes (assuming that the sampling rate is appropriately chosen).

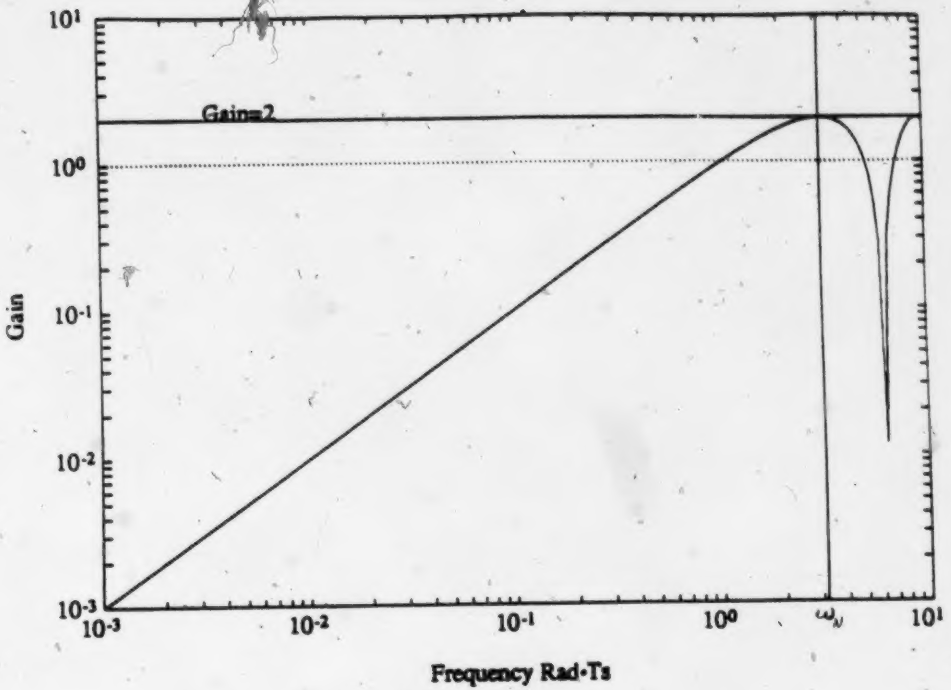

Figure 4.2a Bode Plot of the Differencing Operator Δ

Examination of the cumulative spectral density of a differenced white noise signal clearly demonstrates the emphasis placed on the high frequency bandwidths of a signal. The cumulative spectral density of a signal $\nu(t)$ is calculated by integrating over the spectral density of the output denoted by $\Omega(\omega)$. For differenced white noise it is given by:

$$\Omega(\omega) = \frac{1}{2\pi}(1-z)(1-z^{-1}) = \frac{1}{2\pi}(2 - e^{i\omega T_s} - e^{-i\omega T_s}) \qquad (4.9)$$

The cumulative spectral density is given as a function of frequency ω,

$\omega \in [0,2\pi]$ (Astrom and Wittenmark, 1984) as:

$$P(\omega) = \frac{1}{2\pi i}\int_0^{\omega}(2 - e^{i\zeta T_s} - e^{-i\zeta T_s})e^{-i\zeta T_s}\,d(e^{i\zeta}) \tag{4.10}$$

$$= \omega/\pi - \sin(\omega)$$

which is plotted in Figure 4.2b.

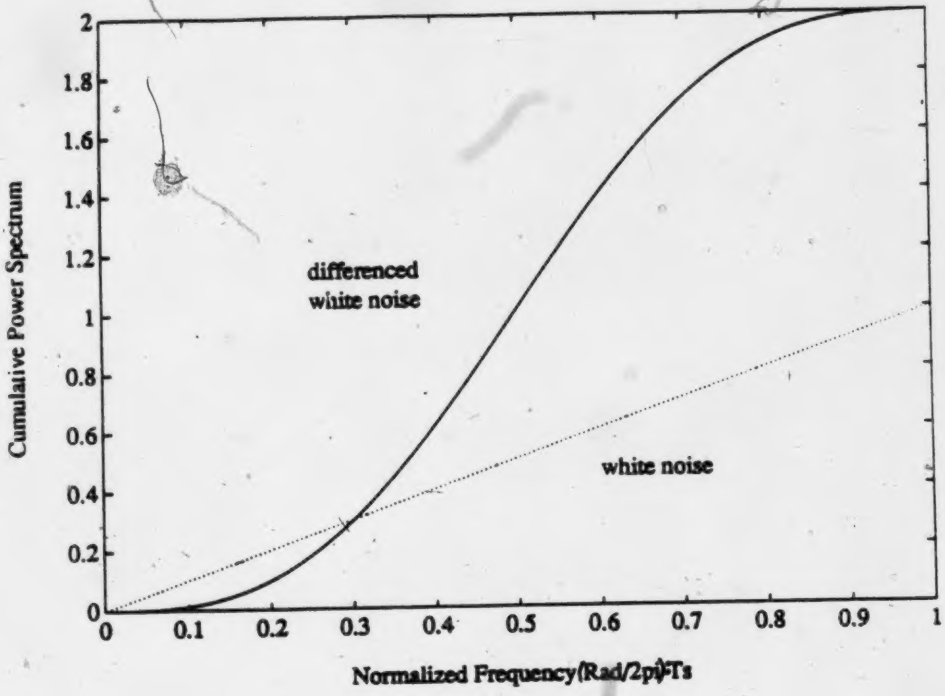

Figure 4.2b Cumulative Spectral Density of a Differenced White Noise Signal

Note that the power gain, given by the integral taken over the complete spectrum, is equal to 2:

$$\frac{E\{(\Delta\xi(t))^2\}}{E\{\xi(t)^2\}} = \frac{1}{2\pi i}\int_{-\pi}^{\pi}(2 - e^{i\omega T_s} - e^{-i\omega T_s})e^{-i\omega T_s}d(e^{i\omega}) = 2$$

A simple approach to the high frequency problems introduced by differencing is low pass filtering of the data. A filtering scheme was suggested for the ARIMAX model (Tuffs, 1985) in which a "tailoring" filter $\frac{1}{T(q^{-1})}$ was chosen to dampen the effects of sudden disturbances by low pass filtering. The polynomial $T(q^{-1})$ is monic and the filter is strictly stable (has all poles within the unit circle of the z^{-1} plane). The filtered incremental data variables u' and y' are given as:

$$y'(t) = \frac{1 - q^{-1}}{T(q^{-1})}y(t) \tag{4.11}$$

$$u'(t) = \frac{1 - q^{-1}}{T(q^{-1})}u(t) \tag{4.12}$$

and the filtered ARIMAX model representation was written as:

$$A(q^{-1})\frac{\Delta y(t)}{T(q^{-1})} = B(q^{-1})\frac{\Delta u(t)}{T(q^{-1})} + \frac{C(q^{-1})}{T(q^{-1})}\xi(t) \tag{4.13}$$

In the work of Tuffs (1985), the polynomial $T(q^{-1})$ was a user specified constant polynomial based on *a priori* knowledge of the disturbance characteristics. The choice of the filter structure and parameters introduced more tuning knobs for the designer to deal with. An identical filtering scheme has been described (Ljung and Soderstrom, 1983) based on the objective of modifying or "frequency shaping" of the prediction error cost function, in order to place less emphasis on frequency bandwidths which contain less useful information.

The $T(q^{-1})$ filter should be chosen as an *a priori* estimate of the $C(q^{-1})$ filter which best describes the true system. Application of this value of $T(q^{-1})$ will filter the data by an approximate inverse of the disturbance model, thus the resulting filtered disturbance will be approximately white.

4.4 Interpretation of the T Filter

A physically meaningful interpretation may be made of the T-filtering scheme given by equations (4.11) and (4.12). Consider the case where $T(q^{-1})$ is chosen as first order:

$$y'(t) = \frac{1 - q^{-1}}{1 + t_1 q^{-1}}\, y(t) \qquad (4.14)$$

$$u'(t) = \frac{1 - q^{-1}}{1 + t_1 q^{-1}}\, u(t) \qquad (4.15)$$

Applying filters of this form is equivalent to subtracting the moving average means from the measured data as given by equations (2.17):

$$u'(t) = u_m(t) - \underline{u}(t) \qquad (4.16)$$

$$y'(t) = y_m(t) - \underline{y}(t) \tag{4.17}$$

with the means $\underline{y}$ and $\underline{u}$ given by exponential moving averages:

$$\underline{u}(t) = (1-\alpha)u_m(t-1) - \alpha\underline{u}(t-1) \tag{4.18}$$

$$\underline{y}(t) = (1-\alpha)y_m(t-1) + \alpha\underline{y}(t-1) \tag{4.19}$$

The moving average weighting coefficient α is clearly related to the t_1 parameter as:

$$\alpha = -t_1 \tag{4.20}$$

The f[illegible]red data values u' and y' may be described as "mean deviational" in a sense, with the "mean sample length" (MSL) (a measure of the number of data points over which the exponential moving average is taken) given by :

$$\text{MSL} = \frac{\alpha}{1-\alpha} = \frac{-t_1}{1+t_1} \tag{4.21}$$

The frequency response of the mean deviational filter with $t_1 = -0.8$ is plotted in Figure 4.3, with the differencing filter (corresponding to $t_1=0.0$) also plotted for comparison.

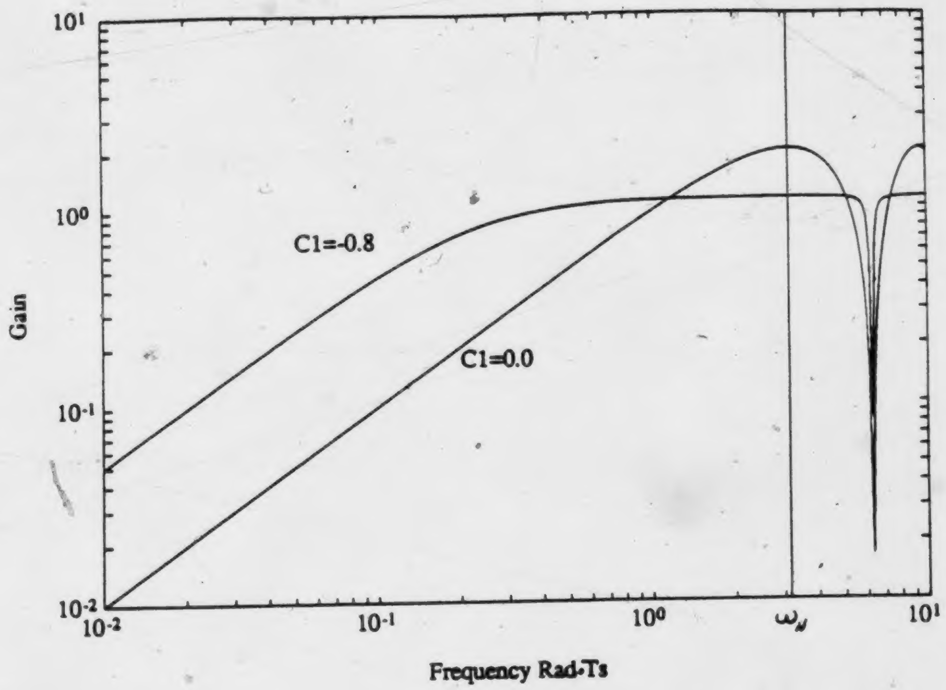

Figure 4.3 Bode Plot of the Mean Deviational Filter

The benefit of the T filter is clearly apparent, since the high frequency amplification is significantly reduced, and the gain is nearly unity in the useful bandwidth $\omega/T_s \in (10^{-1},\pi)$.

Another interpretation of the $T(q^{-1})$ filter and its application to the ARIMAX model is seen in its equivalence to Kalman filtering, as explained in the following section.

4.5 Kalman Filter Interpretation of the ARIMAX Structure

In Chapter 2 it was explained that linear dynamic models are properly expressed in terms of deviations from steady state (eg. equations (2.17a) and (2.17b)). Several approaches were presented for estimating the steady states $\underline{u}$ and $\underline{y}$. The most useful for control purposes are those which lump the steady states into the disturbance model as a non-zero displacement term μ. The integrating noise models consider the closure term to be a non-stationary non-zero mean component of the disturbance whose behavior is described as a Brownian motion or Weiner process (Åström and Wittenmark, 1984). Such models use an internal estimate of μ which is not calculated explicitly but is an important state variable which allows the model to be used as a predictor. The quality of predictions available from a model is clearly dependent on the accuracy of the estimate of μ, even if it does not appear explicitly in the model equation.

An optimal method for the estimation of μ is derived from the application of a Kalman filter to the state equation describing its motion. For the integrating noise models the state equation describing the Brownian motion behavior of the closure term μ is given by:

$$x(t+1) = x(t) + \xi_1(t) \tag{4.22}$$

where $\xi_1(t)$ is a zero mean random variable with variance σ_1^2. The observer equation is written as:

$$z(t) = x(t) + \xi_2(t) \tag{4.23}$$

where $\xi_2(t)$ is random zero mean signal with variance σ_2^2, and z(t) is an observed variable. The Kalman filter estimation equations are as follows:

$$\hat{x}(t+1|t) = \hat{x}(t|t-1) + K(t)\left[z(t) - \hat{x}(t|t-1)\right] \tag{4.24}$$

$$P(t+1) = P(t) + \sigma_1^2 - P(t)\left[\sigma_2^2 + P(t)\right]^{-1} P(t) \tag{4.25}$$

$$K(t) = P(t)\left[\sigma_2^2 + P(t)\right]^{-1} \tag{4.26}$$

If the initial estimate of x(t) is close to the true value then the time varying nature of the equations may be neglected by taking P(t+1)=P(t) in equation (4.25). The resulting steady-state Kalman gain simplifies to:

$$K_{ss} = \frac{-\sigma_1^2 \pm \sigma_1\sqrt{\sigma_1^2 + 4\sigma_2^2}}{2\sigma_2^2} \tag{4.27}$$

The steady state estimation equation is given by substituting K_{ss} for the Kalman gain in equation (4.24):

$$\hat{x}(t+1|t) = \hat{x}(t|t-1) + K_{ss}\left[z(t) - \hat{x}(t|t-1)\right] \tag{4.28}$$

Consider an ARX process in terms of the measured variables $y_m(t)$ and $u_m(t)$:

$$A(q^{-1})y_m(t) = q^{-d}B(q^{-1})u_m(t) + \mu(t) + \xi(t) \qquad (4.29)$$

with the closure term given by a Brownian motion process:

$$\mu(t) = \frac{1}{\Delta}\xi_2(t) \qquad (4.30)$$

For this case the observed variable z(t) is the estimate of μ given in terms of the positional measurements $y_m(t)$ and $u_m(t)$ and the model parameters:

$$z(t) = A(q^{-1})y_m(t) - q^{-d}B(q^{-1})u_m(t) \qquad (4.31)$$

The estimate of the state $\hat{x}(t)$ in equation (4.28) corresponds to the estimate of μ. Substituting the right hand side of equation (4.31) into (4.28) for z(t) gives:

$$\hat{\mu}(t+1) = \hat{\mu}(t) + K_{ss}\left[(A(q^{-1})y_m(t) - q^{-d}B(q^{-1})u_m(t)) - \hat{\mu}(t)\right]$$

$$= (1-K_{ss})\mu(t) + K_{ss}(A(q^{-1})y_m(t) - q^{-d}B(q^{-1})u_m(t))$$

$$= \frac{K_{ss}}{1-(1-K_{ss})q^{-1}}(A(q^{-1})y_m(t) - q^{-d}B(q^{-1})u_m(t)) \qquad (4.32)$$

Writing the estimate (4.32) for $\mu(t)$ and substituting in the model equation (4.29) gives:

$$A(q^{-1})y_m(t) = q^{-d}B(q^{-1})u_m(t)$$

$$+ \left[\frac{K_{ss}}{1-(1-K_{ss})q^{-1}}(A(q^{-1})y_m(t)-q^{-d}B(q^{-1})u_m(t))\right] + \xi(t) \tag{4.33}$$

which reduces to:

$$A(q^{-1})\Delta y_m = q^{-d}B(q^{-1})\Delta u_m(t) + (1 + (K_{ss}-1)q^{-1})\xi(t) \tag{4.34}$$

This is equivalent to the ARIMAX model with $C(q^{-1})$ given by:

$$C(q^{-1}) = 1 + (K_{ss}-1)q^{-1} \tag{4.35}$$

hence the value of c_1 corresponds to $(K_{ss}-1)$, and its value is given by:

$$c_1 = K_{ss}-1 = \frac{-\sigma_1^2 \pm \sigma_1\sqrt{\sigma_1^2 + 4\sigma_2^2}}{2\sigma_2^2} - 1$$

$$= -1 - \frac{\sigma_1^2}{2\sigma_2^2} + \frac{\sigma_1}{2\sigma_2^2}\sqrt{4\sigma_2^2 + \sigma_1^2} \tag{4.36}$$

This result is identical to equation (4.7) which was derived as a linear combination of the stationary and non-stationary disturbances. The ARIMAX structure is hence equivalent to estimating the closure term μ by a Kalman filter. Using a transfer function such as the ARIMAX model however, would appear to have an advantage over the use of Kalman filter since it does not require direct knowledge of σ_1 and σ_2. If the parameter c_1 is known or can be directly estimated then results approaching or equivalent to the exact Kalman gain predictions may be obtained. The recursive methods for estimating c_1 in the ARIMA structure such as RML and ELS may be interpreted as direct methods for estimating the Kalman filter gain, and hence indirect methods of estimating σ_1 and σ_2. The following numerical example provides a practical demonstration of this equivalence.

The following disturbance process was simulated:

$$y(t) = \frac{\xi_1}{\Delta} + \xi_2 \tag{4.37}$$

with the variances σ_1^2 and σ_2^2 chosen to span a range of conditions as summarised in Table 4.1. The true or best values of the parameter c_1 were calculated from σ_1^2 and σ_2^2 using equation (4.36) are also presented in Table 4.1.

Table 4.1 Experimental Conditions for Simulation Example

Run #	σ_1^2	σ_2^2	c_1(calculated)
1	0.1	0.01	-0.08392
2	0.05	0.05	-0.38297
3	0.01	0.2	-0.80000

The system was modeled as a simplified ARIMA process, given by:

$$y(t) = \frac{1 + \hat{c}_1 q^{-1}}{1 - q^{-1}} \xi\ t) \tag{4.38}$$

The parameter $\hat{c}_1$ was estimated by two methods corresponding to the ELS and RML algorithms as described in Chapter 3. These were implemented by choosing the regressor ϕ and the parameter vector θ as scalar values:

$$\phi(t) = \left[e(t-1|t-1) \right] \quad \hat{\theta}(t) = \left[\hat{c}_1 \right]$$

with the prediction error given by:

$$e(t|t) = \Delta y(t) - \hat{\theta}(t|t)\phi(t)$$

The ELS method was implemented using $\phi(t)$ as the gradient of the prediction error:

$$P(t) = P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{1 + \phi(t)P(t-1)\phi^T(t)}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\phi(t)e(t|t-1)}{1 + \phi(t)P(t-1)\phi^T(t)}$$

The RML method was implemented using a "true" gradient given by:

$$\psi(t) = \frac{1}{\hat{C}(q^{-1})} \phi(t) = \phi(t) - \hat{c}_1 \psi(t-1)$$

For this system the closure term μ is an independent function of time because it is the non-zero mean component of the system, and it is time varying. Its value is given as:

$$\mu(t) = \frac{1}{\Delta}\,\xi_1(t) \tag{4.39}$$

The estimate of $\mu(t)$ was calculated by:

$$\hat{\mu}(t) = \frac{(1+c_1)}{1+c_1q^{-1}}\,y(t) \tag{4.40}$$

and a comparison is made between three different choices for the value of c_1:

1) The value calculated from (4.36) with the values of σ_1^2 and σ_2^2 known (equivalent to the Kalman filter)

2) The estimate $\hat{c}_1(t)$ from the application of RML to the model given by equation (4.38).

3) The estimate $\hat{c}_1(t)$ from the application of ELS to the same model.

The comparison is made by examining the integral of the squared error in the estimation of $\mu(t)$ given by:

$$V = \frac{1}{N}\sum_{t=1}^{N}[\hat{\mu}(t)-\mu(t)]^2 \tag{4.41}$$

The simulations were programmed in Microsoft FORTRAN-77 and run on an IBM AT computer. The source code for the program is included in Appendix B. Each case was run for 1000 intervals and the results are presented in Table 4.2. The reported values of $\hat{c}_1$ for the RML and ELS cases are the estimates at t=1000.

Table 4.2 Comparison of Transfer Function Estimation versus Kalman Filtering Estimation of the Closure term

	Kalman Filter Method		ELS Method		RML Method	
Run #	V	c_1	V	$\hat{c}_1$	V	$\hat{c}_1$
1	0.00943	-0.0839	0.01017	-0.0617	0.01039	-0.0620
2	0.03243	-0.3820	0.03251	-0.3583	0.03272	-0.3231
3	0.04539	-0.8000	0.04521	-0.7746	0.05742	-0.6149

The parameter trajectories for each run are plotted in Figures 4.4a,b,c respectively. Since an exact convergence proof has been demonstrated for both the RML and ELS methods under these conditions (Ljung and Soderstrom, 1983) the estimate of c_1 should asymptotically reach the true value of c_1, if the noise sequences were perfectly white.

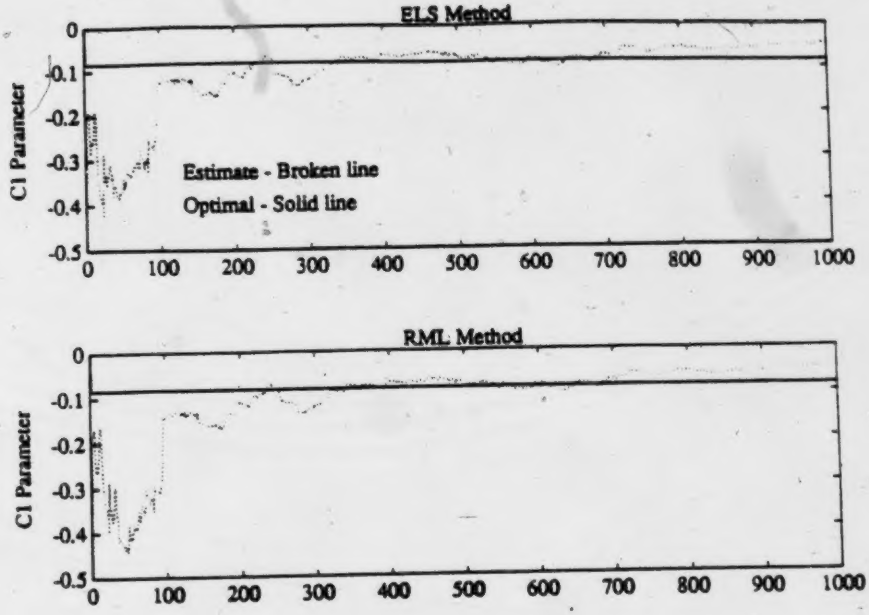

Figure 4.4a Parameter Trajectories Run #1

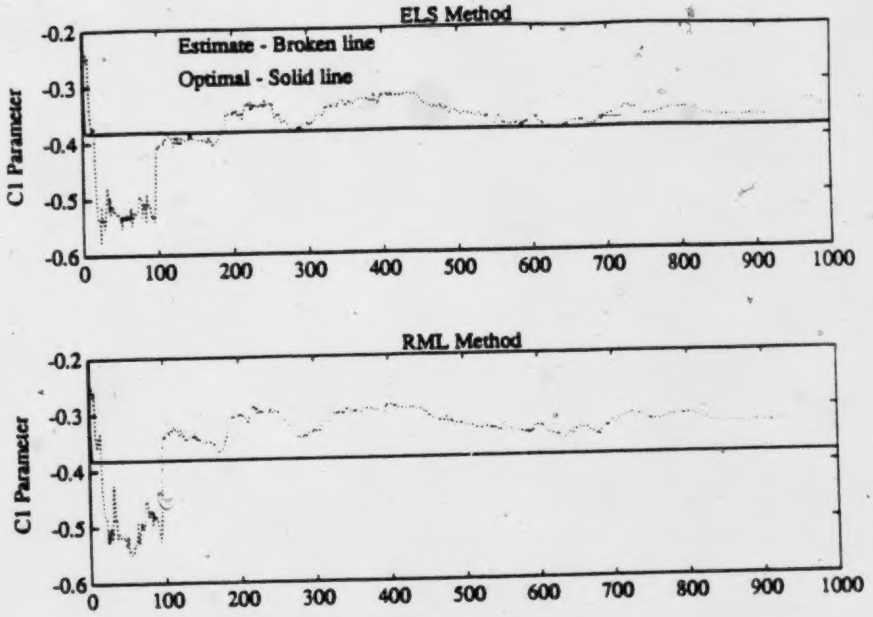

Figure 4.4b Parameter Trajectories Run #2

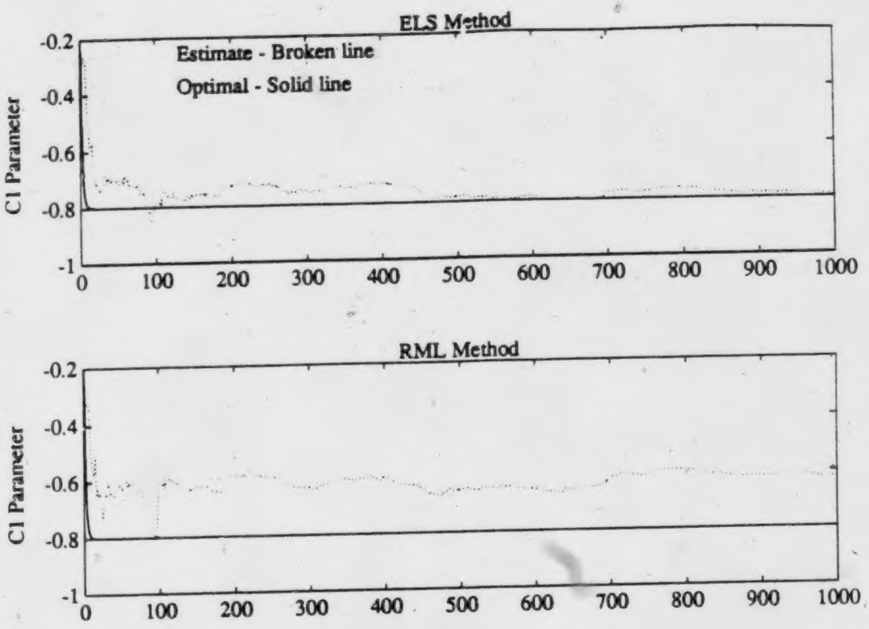

Figure 4.4c Parameter Trajectories Run #3

In all three cases the estimated value of c_1 after 1000 intervals approached a constant value, however no significant improvement was noted for longer runs. It is postulated that round-off effects of the computer prevent exact convergence. It is interesting to note that the RML algorithm gives poorer estimates of c_1 (compared to the ELS method) as the actual value of c_1 becomes closer to -1. This phenomenon may also be attributable to round-off effects, since the RML method involves a data filtering step at each interval that is avoided in ELS method. It is recommended that this be examined as a topic for future research.

The estimates of $\mu(t)$ obtained by modeling the system as an ARIMA process are close to the optimal Kalman filter estimation of $\mu(t)$ in all three cases examined. The RML method was generally slower to converge and not as accurate in the final estimation of the c_1 parameter than the ELS method. This example demonstrates that the estimate of $\mu(t)$ given by the ARIMA model is nearly equivalent to the optimal estimate of $\mu(t)$ given by the Kalman filter with the noise properties exactly known. The results presented support the use of the ELS method for the estimation of the Kalman gain in situations where the noise properties are unknown or subject to slow variations in time.

4.6 Summary

Chapter 4 examines some theoretical and practical interpretations of using an integrating (or Brownian motion) noise model to describe disturbances. The practice of differencing the input and output measurements from a system in order to fit the data to a discrete transfer function model is based on an underlying assumption that a component of the disturbance may be described by a Brownian motion model.

It is demonstrated that a basic admissibility condition on a first order C polynomial is that the value of c_1 must lie in the range [0,-1]. This is consistent with the postulate that the ARIMAX model results from an unfiltered Brownian motion process plus a filtered white noise signal. A generalised admissibility criterion is derived based on the postulate.

It is well known that simple differencing of the data leads to problems in the presence of noise. Examination of the frequency domain properties of the differencing filter reveals that it places undue emphasis on data in the higher frequency bandwidths, (close to the Nyquist frequency), and that the overall power spectrum is doubled. This is a significant problem in identification because typically the model is a reduced order representation of the system, and in these cases it is more important to closely model the low frequency bandwidths. Furthermore the higher frequency bandwidths generally have a higher proportion of noise and random effects hence the data is less reliable. These problems may be reduced or eliminated by the addition of a low pass T filter to the differencing operator, which in the first order case reduces to a "mean deviational" filter as presented in Chapter 2.

In Section 4.5 it was demonstrated that the C polynomial of the ARIMAX structure is mathematically equivalent to a Kalman filter for tracking the value of the displacement term μ. The mathematical equivalence of the equations is proven and a numerical example using the RML and ELS methods was presented to demonstrate the practicality of this result. In the example it is concluded that the ELS method, although less rigorously supported in theory (than the RML method) gives superior performance. Its general use is therefore recommended for applications for which the noise properties are unknown or are subject to slow variations in time.

Chapter 5

Exponential Forgetting Factors

5.1 Introduction

A key problem with the application of least-squares type algorithms for recursive applications is that of estimator "shut-off" due to the continually diminishing update gain. There have been many proposed modifications which offer solutions (a brief review is presented in Shah, 1986), however this subject apparently remains problematic, and a significant barrier to the general acceptance of adaptive control in industry.

Chapter 5 considers three different methods for calculating variable forgetting factors for use in exponential forgetting schemes. All of the methods considered maintain estimator alertness by keeping an arbitrary measure of the P matrix equal to a constant. The methods differ as to which measure is used, and correspondingly, each has at least one "tuning knob" relating to that magnitude.

One of the forgetting factors presented is new and based on the idea of maintaining the determinant of the P matrix equal to a constant. The basic advantage of this approach is that singularity of the P matrix is inherently avoided, and the resulting equations are therefore less susceptible to numerical problems which may be associated with the value of this "tuning knob". Furthermore, if the determinant is held constant then both the largest and smallest eigenvalues of P are bounded by tr(P), therefore the algorithm is easily monitored and constrained.

A key observation by Kulhavy (1987) is that the data vector $\phi(t)$

must contain information which is uniformly distributed over time and the parameter space. If this condition is not fulfilled then application of the exponential forgetting factor results in the loss of a piece of so-far accumulated information, which is not compensated for by gain of new information. In practice this effect is recognised as covariance windup. Kulhavy (1987) concluded that this restriction occurs regardless of the type of variable forgetting factor used, and in theory all methods which use exponential forgetting algorithms will require some mechanism which prevents windup. In practice however, it is found that some forgetting factors go to unity when the system is not uniformly excited. For these the windup phenomenon is avoided in a natural way by causing the forgetting to cease.

A simulation example and an experimental evaluation using process data were performed to compare the three previously mentioned exponential forgetting factor algorithms under conditions of non-informative data, and in the presence of noise. In the simulation example the algorithms were applied to a simple linear system operating under closed loop proportional feedback conditions. This situation represents a particularly difficult problem in recursive identification because the data available from the process are insufficient to identify the parameters, regardless of the duration of the run.

The experimental study was conducted by applying the algorithms to experimental process data which were collected from operation of a pilot scale distillation column under closed loop conditions. The results from the simulation are verified and the practicality of the new constant determinant forgetting factor is demonstrated under realistic conditions.

The confidence bound ellipsoid is presented as a tool for analysis of the numerical properties of the P matrix, and it is used to graphically examine the effect of the forgetting factor on these properties in the simulation example. The ellipsoid provides interesting insight into the geometric properties of the algorithms and is useful for interpreting and comparing the effects of the forgetting factor in adaptive control studies.

5.2 Estimation Shut-off In RLS Methods

A recognised problem of the RLS type methods in adaptive applications is that the $P^{-1}(t)$ matrix defined in equation (3.19) is non-decreasing and grows without bound in time because the rank one, positive semi-definite matrix given by $\alpha_t\phi(t)\phi^T(t)$ is continually added to it:

$$P^{-1}(t) = \sum_{k=1}^{t}\alpha_k\phi(k)\phi^T(k) = \overline{R}(t-1) + \alpha_t\phi(t)\phi^T(t) \tag{3.19}$$

The matrix $P(t)$ will asymptotically decrease with time, and the estimator gain vector given by $P(t)\phi(t)$ diminishes to zero causing the parameter updating to asymptotically shut off or "sleep". This effect can cause the parameter estimates to be biased in two situations:

1) The initial estimate $\hat{\theta}(0)$ is not sufficiently close to θ^*. The $P(t)$ matrix and the update gain $P(t)\phi(t)$ become very small before $\hat{\theta}(t)$ reaches θ^*.

2) The value of θ_0 is time varying, hence the parameter estimates are unable to track changes after $P(t)$ becomes small.

The first problem is solved by choosing $P(0)$ sufficiently large that the initial parameter covariance bound "encloses" the optimal estimate θ^*. The P(0) matrix must be chosen sufficiently large to satisfy the following relation:

$$\tilde{\theta}^T(0)P^{-1}(0)\tilde{\theta}(0) \le 1/\sigma^2 \tag{5.1}$$

A common approach to the second problem is to inflate the P matrix in some arbitrary manner, either periodically or at each interval. Such methods effectively maintain the "alertness" of the least squares algorithm by artificially increasing the estimate of the parameter covariance. This process is generally known as forgetting, since it increases the apparent uncertainty about the parameter estimates.

The following section describes a geometric interpretation of the P matrix which allows some interpretation of the methods of forgetting.

5.3 A Geometric Interpretation of the P Matrix

The matrix P(t) is related to the covariance of the joint distribution of the parameter vector estimates, $\hat{\theta}(t)$ by a Lyapunov-type function:

$$(\theta-\hat{\theta}(N))^T P^{-1}(N)(\theta-\hat{\theta}(N)) \in \chi^2(N-n,c)/N \tag{5.2}$$

This equation states that the probability that:

$$(\theta-\hat{\theta}(N))^T P^{-1}(N)(\theta-\hat{\theta}(N)) \ge c/N \tag{5.3}$$

is $\chi^2(N-n,c)$, the c-level of the χ^2 distribution with N-n degrees of freedom. The expressions given by values of c in (5.3) define ellipsoids in $\mathbb{R}^n$ centred at $\hat{\theta}(N)$, whose shapes are defined by P(N). The lengths of the semi axes of the ellipsoid are proportional to the square root of the eigenvalues of P(N), or equivalently the eigenvalues of the square root factorisation of P(N). The axes of the ellipsoid correspond to the eigenvectors of the P matrix. Recall that the determinant of the P matrix is the product of its eigenvalues hence $|P|$ is proportional to the square of the area of the ellipse. The condition number (given by the ratio of the largest to the smallest eigenvalue) is geometrically indicated by the degree of elongation or distortion of the ellipse from a circular shape.

Figure 5.1 depicts a covariance bound ellipse in two dimensions corresponding to the values:

$$P = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}; \; c/N = 1.0$$

The length of semi-axes a and b correspond to the square roots of the eigenvalues of P, $\ell_a = \sqrt{3}$ and $\ell_b = 1$. The rotation of the axes is due to the eigenvectors $\begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$, and $\begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}$.

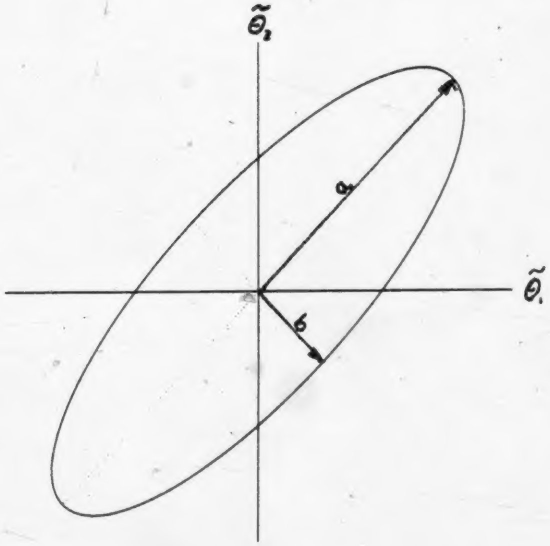

Figure 5.1 Confidence Bound in $\mathbb{R}^2$

A series of plots of the confidence bound ellipses provide a graphical "view" of the P matrix, useful for analysing its evolution during an experiment. In the following sections this tool is used to graphically compare the behavior of different forgetting factor algorithms. The ellipse interpretation is also useful in understanding the application of a recursive least squares type algorithm to the problem of the bounded noise description as will be described in Chapter 7.

5.4 Exponential Forgetting Factors

A simple method for inflating the P matrix in a recursive fashion is to introduce a forgetting factor λ into the covariance update equation which reduces the weighting on old data observations in an exponential manner. The value of λ is usually taken in the interval 0.9-1.0. The effect of the forgetting factor is that P(t) and hence the gain $P(t)\phi(t)$ will not tend to zero and the algorithm will remain alert to track changing dynamics.

The RLS equations with a forgetting factor may be derived from weighted least squares by taking the weighting sequence α_i to be a geometric series with an exponential profile. If λ is chosen as a time dependent variable then the weighting sequence is given in terms of $\lambda(j)$, $0<j\leq t$ as:

$$\alpha_t = 1; \quad \alpha_{t-1} = \lambda(t); \quad \alpha_{t-2} = \lambda(t)\lambda(t-1); \ \ldots$$

$$\alpha_i = \prod_{j=i+1}^{t} \lambda(j) \ ; \ i>1 \qquad (5.4)$$

The recursive update of $P^{-1}(t)$ will be given as:

$$P^{-1}(t) = \left[\prod_{i=1}^{t} \lambda(i)\right] P^{-1}(0) + \sum_{i=1}^{t} \left[\prod_{j=i+1}^{t} \lambda(j)\right] \phi(i)\phi^T(i)$$

$$= \lambda(t)\left[\prod_{i=1}^{t-1} \lambda(i)P^{-1}(0) + \sum_{i=1}^{t-1}\left[\prod_{j=i+1}^{t-1} \lambda(j)\right]\phi(i)\phi^T(i)\right] + \phi(t)\phi^T(t)$$

$$= \lambda(t)P^{-1}(t-1) + \phi(t)\phi^T(t) \tag{5.5}$$

Writing the update equations in terms of the P(t) matrix and applying the matrix inversion lemma as equation (3.21) gives:

$$P(t) = \left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{\lambda(t) + \phi^T(t)P(t-1)\phi(t)}\right]\frac{1}{\lambda(t)} \tag{5.6}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\phi(t)\epsilon(t)}{\lambda(t)+\phi^T(t)P(t-1)\phi(t)} \tag{5.7}$$

Exponential forgetting may be derived from ordinary LS in a slightly different manner by assuming that the forgetting step should occur between sample intervals. In this approach the covariance update is thus performed in two steps, necessitating the use of more explicit time subscripts. The resulting algorithm is given as follows:

1) The forgetting step which occurs between sample intervals gives the update $P(t|t-1) \rightarrow P(t|t)$ as:

$$P(t|t) = P(t|t-1)/\lambda(t) \tag{5.8}$$

2) With each new data observation the update $P(t|t) \rightarrow P(t+1|t)$ is given as:

$$P^{-1}(t+1|t) = P^{-1}(t|t) + \phi(t)\phi^T(t) \tag{5.9}$$

$$\hat{\theta}(t+1|t) = \hat{\theta}(t|t-1) + P(t+1|t)\phi(t)\epsilon(t) \tag{5.10}$$

Applying the matrix inversion lemma, equations (5.9) and (5.10) may be written in terms of P(t|t) as:

$$P(t+1|t) = P(t|t) + \frac{P(t|t)\phi(t)\phi^T(t)P(t|t)}{1 + \phi^T(t)P(t|t)\phi(t)} \tag{5.11}$$

$$\hat{\theta}(t+1|t) = \hat{\theta}(t|t-1) + \frac{P(t|t)\phi(t)\epsilon(t)}{1 + \phi^T(t)P(t|t)\phi(t)} \tag{5.12}$$

In practice it is found that equations (5.11) and (5.12) behave similarly to those derived from weighted least squares (equations (5.6) and (5.7).

An interesting interpretation of the exponential forgetting factor may be made by examining the effect of the covariance update equations on the confidence bound ellipsoid. Figure 5.2 provides a graphical picture of this. The data update step causes one dimension of the ellipse to contract, in the direction of the vector P(t-1)ϕ(t), while the exponential forgetting step causes the ellipse to be proportionately expanded in all directions.

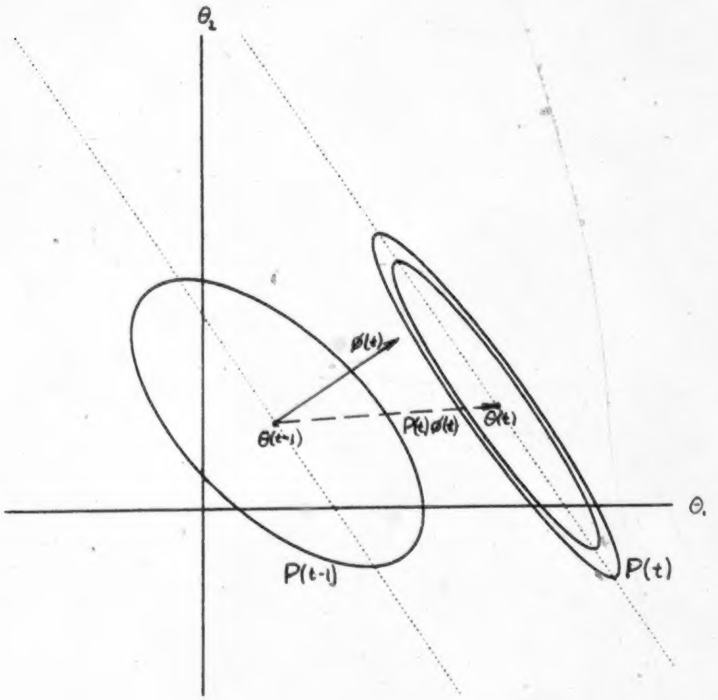

Figure 5.2 Effect of Exponential Forgetting on the Confidence Bound Ellipse

A common problem with using a forgetting factor is that as the term $\phi^T(t)P(t-1)\phi(t)$ becomes small the update equation (5.6) reduces to:

$$P(t) = \frac{1}{\lambda}P(t-1) \tag{5.13}$$

Under these conditions the P(t) matrix will grow without bound, a condition which is commonly termed covariance windup (Wittenmark and Åström, 1984). The parameter update gain vector $P(t)\phi(t)$ will also grow causing the parameter estimates to drift excessively. In adaptive control schemes the parameter drift effect can lead to a phenomenon known as bursting

(Wittenmark and Åström, 1984), which occurs when the parameter drift causes the closed loop to move towards a condition of instability. When this occurs, the system may react wildly to slight perturbations, hence the term bursting. A large advantage of projection algorithms for time varying parameter identification applications is that the update gain is constant, and no covariance matrix is used. Such an algorithm is "robust" in the sense that the problems of shutoff and covariance windup are inherently avoided, although this valuable property is generally gained at the expense of convergence rate (Goodwin and Sin, 1983). The idea of maintaining a constant gain in a least squares type method was suggested by Landau and Lozano (1979). In this work it was proposed that the forgetting factor $\lambda(t)$ be chosen to maintain some measure of the magnitude of $P(t)$ equal to a constant. The directional properties of the resulting algorithms should approximate those of least squares (hence one would expect rapid convergence if the estimate is near the convergence point) while the gain vector will be non-diminishing and have an upper bound. The following sections outline different approaches to the idea of choosing $\lambda(t)$ to maintain a measure of the P matrix equal to a constant.

5.5 Constant Information Forgetting Factor

The variable forgetting factor due to Fortescue *et al.* (1981) is computed with the objective of maintaining a constant measure of information in the parameter estimates. The information measure considered is the cumulative sum of the residuals, computed by:

$$\Psi_t = \sum_{i=1}^{t} \frac{\alpha_{i,t}}{\sigma^2}(y_i - \hat{\theta}^T(i)\phi(i))^2 \qquad (5.14)$$

The weighting sequence α_i is chosen to artificially increase the variance of old measurements as they age. If the true value of σ^2 is known then the expected value of the information measure Ψ_t will simply be t, and Ψ_t may be interpreted as an effective memory length. The weighting factor α_i is related to the forgetting factor as in equation (5.4):

$$\alpha_{t,t} = 1; \quad \alpha_{t-1,t} = \lambda(t); \quad \alpha_{t-2,t} = \lambda(t)\lambda(t-1) \; \ldots$$

$$\alpha_{i,t} = \prod_{j=i+1}^{t} \lambda(j) \; ; \; i>1 \qquad (5.4)$$

By neglecting the contribution of initial information a recursive expression may be derived for Ψ_t from equation (5.16) in terms of the forgetting factor:

$$\Psi_t = \lambda(t)\left[\Psi_{t-1} + \frac{\epsilon^2(t)/r}{(\lambda(t)+G(t))}\right] \qquad (5.15)$$

with: $G(t) = \phi^T(t)P(t-1)\phi(t)$

and r chosen as an estimate of σ^2.

The variable forgetting factor $\lambda(t)$ may be calculated from equation (5.14) in terms of the prediction error and G(t) to maintain the information measure Ψ at a constant by setting $\Psi_t = \Psi_{t-1} = \Psi_0$ in equation (5.15):

$$\lambda(t) = 1 - \frac{\lambda(t)\epsilon^2(t)}{(\lambda(t) + G(t))r\Psi_0} \qquad (5.16)$$

Since $G(t)>0$, this equation has a strictly positive root which satisfies $\lambda(t)\in(0,1]$ given by:

$$\lambda(t) = \left[m_t + \sqrt{m_t^2 + 4G(t)}\right]/2 \; ; \qquad (5.17)$$

with: $m_t = 1 - G(t) - \epsilon^2(t)/(r\Psi_0)$

In practice the actual value of σ^2 is unknown, but if the effects of the initial conditions are ignored then the true covariance matrix of the parameter estimates is given by the P(t) matrix scaled by the variance of the innovations sequence σ^2:

$$E\{(\hat{\theta}(t)-\theta^*)(\hat{\theta}(t)-\theta^*)^T\} = \lim_{t\rightarrow N} \{P(t)\sigma^2\} \qquad (5.18)$$

The estimate of the noise variance r may be assumed equal to unity and omitted from equation (5.14) without loss of generality.

Fortescue *et al.* (1981) observed that the second term of equation (5.16) will be small, causing $\lambda(t)$ to approach unity in any of the following cases:

1. The process is quiet i.e. $\phi(t)$ is small hence $\epsilon(t)$ is small.

2. The parameter estimates $\hat{\theta}(t)$ are close to the correct values, hence $\epsilon(t)$ is small.

3. The uncertainty in the estimates $\hat{\theta}(t)$ given by P(t-1) is large, hence G(t) is large.

4. The effective memory length Ψ_0 is large.

Values of $\lambda(t)$ significantly less than unity occur however when the error $\epsilon(t)$ is large and G(t) is small. During long periods of steady operation (during which the process is not persistently excited) the data may be only due to noise, hence it is likely that the information content of the P matrix will be due to only noise, allowing all the important information to be forgotten. It is therefore possible for the covariance matrix to become large, causing the parameters to drift and potentially lead to the bursting phenomenon.

The inclusion of the prediction error in the calculation of the forgetting factor (Fortescue *et al.* 1981) is based on an underlying assumption that inflating the P matrix will increase parameter adaptation, which will in turn reduce prediction errors. This presumes that a stable feedback loop must exist between the prediction errors and the magnitude of P. If this loop does not exist or it is not stable then the P matrix will inflate without bound, or until a bursting phenomenon occurs.

An on/off criterion which shuts off the covariance matrix update when the trace of P exceeds an upper bound was proposed to protect the algorithm from the problem of matrix inflation (Shah, 1986). In this algorithm the matrix update step is given by:

$$P(t) = \begin{cases} M(t) & \mathrm{tr}(M(t)) \leq \kappa \\ P(t-1) & \mathrm{tr}(M(t)) > \kappa \end{cases}$$

with:

$$M(t) = \left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{1 + \phi^T(t)P(t-1)\phi(t)}\right]\frac{1}{\lambda(t)} \tag{5.19}$$

and κ is chosen as a user specified upper bound on tr(P). This effectively shuts off the matrix update when the system is not uniformly excited, however a key problem with it, as with other methods which arbitrarily skip the update of P is that it does not allow "learning" to occur in other directions for which there may be useful information.

5.6 Constant Trace Forgetting Factor

A practical strategy for constraining the magnitude of P is to chose $\lambda(t)$ in order to maintain tr(P) equal to a constant (Landau and Lozano, 1979). The trace of a matrix is the sum of its diagonal elements and is thus a useful measure of the gain of the estimator as it also corresponds to the sum of eigenvalues of the P matrix. A formula for such a forgetting factor has been derived (Sripada and Fisher, 1987) as:

$$\lambda(t) = 1 - \left[\beta(t) - \left\{\beta(t)^2 - \frac{4\|P(t)\phi(t)\|^2}{tr\{P(t-1)\}}\right\}^{1/2}\right]/2 \tag{5.20}$$

$$\beta(t) = 1 + G(t)$$

The basic advantage of the constant trace forgetting factor algorithm is that it effectively maintains estimator alertness while placing an upper bound on the estimator gain. Certain other properties of the matrix are lost however if the data are not persistently exciting. Under these conditions the estimator gain may approach zero in certain directions, (corresponding to one or more eigenvalues of the P matrix approaching zero). This phenomenon may be detected by monitoring another property of the matrix, such as the determinant or the condition number. An on/off criterion based on cond(P_s) has been proposed (Sripada and Fisher, 1987) where P_s is an "optimally scaled" (Noble, B., 1969) version of the P matrix.

The calculation of the forgetting factor by equation (5.20) is numerically robust, since it has been shown (Vien, 1989) that the value of the terms under the square root will always be positive. However the choice of tr(P) is important since it specifies the confidence level which is to be maintained. Choosing the trace too large will result in forgetting factors which are unreasonably small, and choosing it too small will result in forgetting factors equal to unity.

A disadvantage of using tr(P) as a tuning knob is that it is a dimensional quantity, therefore its value will depend on the scaling of the variables in the data vector, which is problem specific.

5.7 Constant Determinant Forgetting Factor

An alternative to using tr(P) as a measure of its magnitude is the determinant, $|P(t)|$. Maintaining a constant determinant has the advantage of providing estimator alertness while inherently avoiding matrix singularity. The constant determinant idea, however does not place an upper or lower bound on the eigenvalues of the P matrix, but rather it forces the product of all the eigenvalues to be equal to a constant. In order to place an upper or lower bound on the eigenvalues it is necessary to monitor and limit some other measure of P, such as tr(P) or cond(P).

A forgetting factor which maintains $|P(t)|=|P(t-1)|$ is computed by setting equal the determinants of both sides of the matrix update equation. An equation for the determinant of P(t) is derived from Theorem 5.1, stated as follows.

Theorem 5.1: Determinant of the Rank One Update

The determinant of the sum of an n×n positive definite matrix A^{-1} multiplied by a scalar α, and the rank one matrix xx^T (x is a vector with column dimension n) is given by the identity:

$$|\alpha A^{-1} + xx^T| = |A^{-1}|\alpha^{n-1}(\alpha + x^T A x) \tag{5.21}$$

□

The proof of Theorem 5.1 is presented in Appendix 1.

Remarks:

Theorem 5.1 applied to the covariance matrix update equations (5.8) - (5.12) gives the value |P(t)| in terms of |P(t-1)| by the formula:

$$|P(t)| = \frac{|P(t-1)|}{\lambda(t)^{n-1}(1 + G(t))} \tag{5.22}$$

Setting |P(t)| = |P(t-1)| gives:

$$\lambda(t)^{n-1}(1 + G(t)) = 1$$

$$\lambda(t) = (1 + G(t))^{-1/(n-1)} \tag{5.23}$$

From equation (5.23) it is clear that the forgetting factor has a solution within the range [0,1] for any value of $\phi(t)$, since $G(t) \geq 0$. The constant determinant algorithm will therefore be free from numerical difficulties associated with the choice of $|P|$, but the computation of $\lambda(t)$ by equation (5.23) may be inaccurate when applied to a computer because it involves an exponential term. As a result some drift in the value of $|P|$ may be expected over long periods. Application of the upper diagonally factored RLS algorithm known as RUD (Bierman, 1977) to the update equation (5.11) allows the determinant to be easily monitored, and hence periodically corrected as necessary. In the RUD algorithm, the P matrix is factorised as $P = U^T \cdot D \cdot U$, where U is an upper triangular matrix with diagonal elements equal to unity, and D is a diagonal matrix. With such a factorisation the value of $|P|$ may be computed directly from D by the relation:

$$|P| = |D| = \prod_{i=1}^{n} D_{ii}$$

Corrections may be made to $|P(t)|$ by suitable adjustment of $\lambda(t)$.

5.8 Rank Deficient Data

It has been observed (Kulhavy, 1987) that an implicit assumption behind the use of the exponential forgetting factor is that the measured signals contain information which is uniformly distributed over the entire parameter space, and is also uniformly distributed in time. Furthermore, if the data are rank deficient (do not span the parameter space) over a period of time, the application of an exponential forgetting factor will cause numerical problems in the calculation of the P matrix. Consider a set of data vectors $\phi(i)$ of dimension n×1 collected from a system which span only n-1 dimensions of the parameter space over some interval $j \leq i \leq k$. Under these conditions a vector x exists such that $x^T\phi(i)=0$. When no forgetting factor is employed the P^{-1} matrix update is given as:

$$P^{-1}(t) = P^{-1}(0) + \sum_{i=j}^{k} \phi(i)\phi^T(i)$$

The x-direction component of the P matrix will remain constant, which can be seen by pre-multiplying both sides by vector x^T, and post-multiplying by x:

$$x^T P^{-1}(t)x = x^T P^{-1}(0)x + x^T \left[\sum_{i=j}^{k} \phi(i)\phi^T(i) \right] x = x^T P^{-1}(0)x \qquad (5.24)$$

The second term on the right hand side of (5.24) is equal to zero, hence the left side will remain constant, equal to the initial value, $x^T P^{-1}(0)x$. If, however, a forgetting factor is employed which takes values less than unity in the interval j→k the relation (5.24) becomes:

$$x^T P^{-1}(t)x =$$

$$\prod_{i=j}^{k} \left[\lambda(i) \right] x^T P^{-1}(0)x + x^T \left[\sum_{i=j}^{k} \prod_{m=i}^{k-1} \left[\lambda(m) \right] \phi(i)\phi^T(i) \right] x \qquad (5.25)$$

Again, the second term on the right hand side of (5.25) is equal to zero. The first term, however, will exponentially tend to zero as the interval j→k becomes large, causing the left hand side to correspondingly diminish toward zero. This implies that at least one eigenvalue of $P^{-1}(t)$ is approaching zero, therefore one eigenvalue of P(t) must become large. This inflation of P(t) in the direction of x is unbounded, and in practice would result in a very large estimator gain in that direction. Large parameter variations would result if the character of the data were to change even slightly.

5.9 Simulation Example

A challenging test of the exponential forgetting factor algorithms results from their application to rank deficient data records over a long period of time. Under such conditions it is interesting to compare how effectively an algorithm prevents the tendency for the P matrix to inflate excessively. In order to test this the previously described forgetting factor algorithms were applied by simulation to the identification of the parameters of the two parameter first order discrete system expressed as:

$$(1 - 0.6q^{-1})y(t) = 0.4u(t-1) + \xi(t) \tag{5.26}$$

The noise term $\xi(t)$ was chosen to be zero mean with variance of 0.1. The input u(t) was generated by proportional feedback with the gain K calculated using the parameters of the estimated model:

$$u(t) = Ky(t)$$

with

$$K = \frac{(1 - \hat{a})}{\hat{b}}$$

These conditions were chosen to generate data with restricted excitation, since proportional feedback will cause rank deficiency. Since only two parameters are being identified and the elements of the data are linearly related by feedback, the data vectors x(t) will be parallel, hence the information has only one directional component.

The identification algorithms were initialised with $P(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\hat{\theta}(0) = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$. The basic algorithm used for updating the parameters and the P matrix is given by equations (5.6) and (5.7).

Figures 5.3a,b, and c are plots of the parameter and forgetting factor trajectories for the three cases. Parameter estimates shown have been normalised by dividing by the true (known) parameter value.

The smallest parameter variation occurs with the constant trace forgetting factor method and the largest with the constant information. In the constant determinant and constant trace cases the value of $\lambda(t)$ asymptotically approaches unity. The constant information forgetting factor however, apparently does not approach unity over the course of the experiment, and a windup effect is evident.

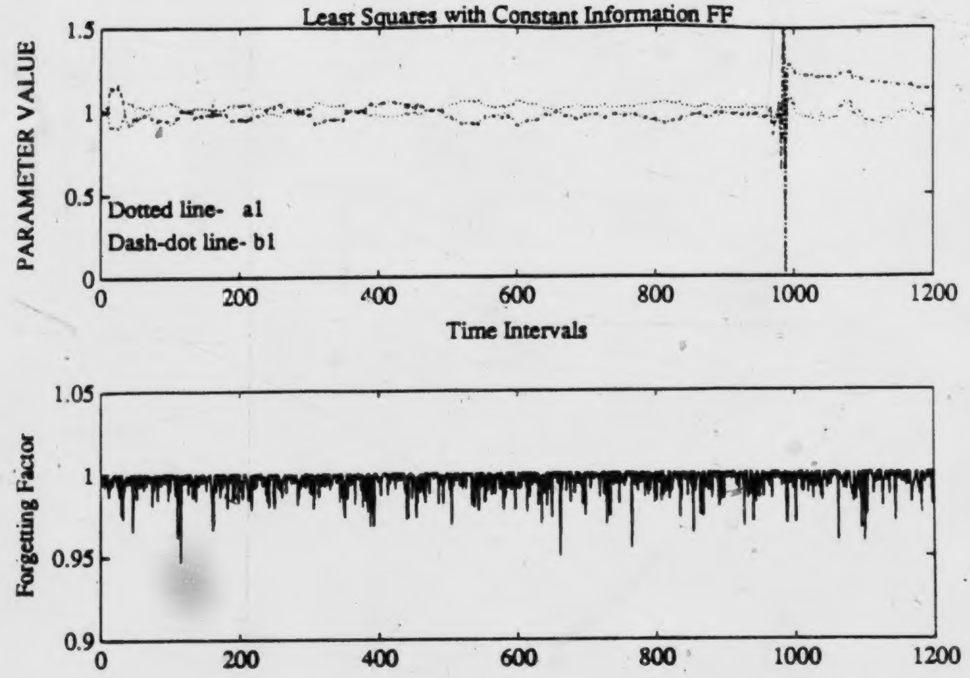

Figure 5.3a Parameter and Forgetting Factor Trajectories: Constant Information Forgetting Factor

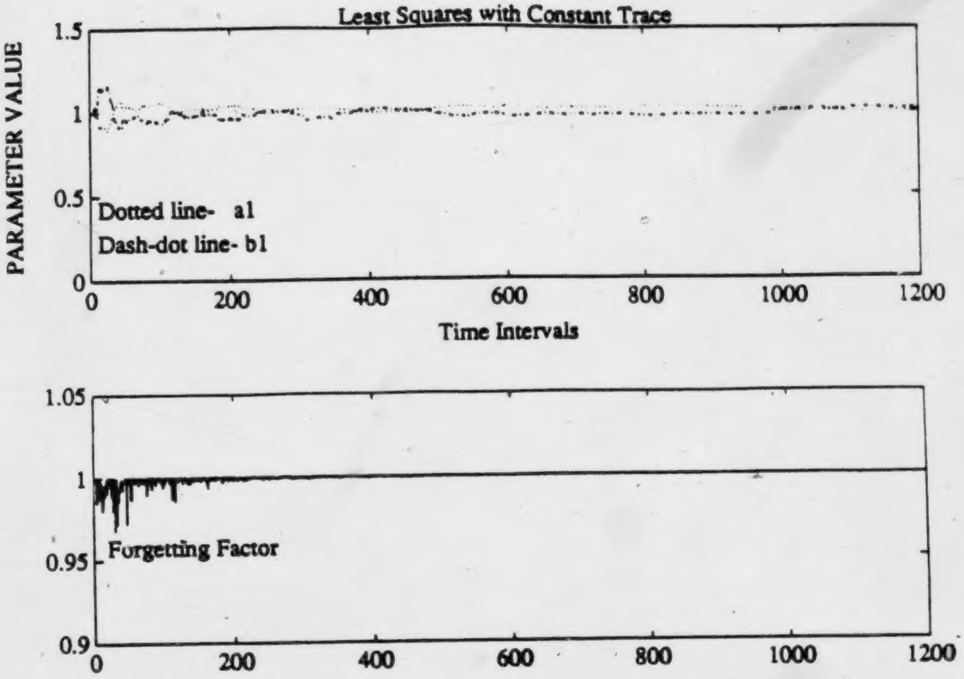

Figure 5.3b Parameter and Forgetting Factor Trajectories: Constant Trace Forgetting Factor

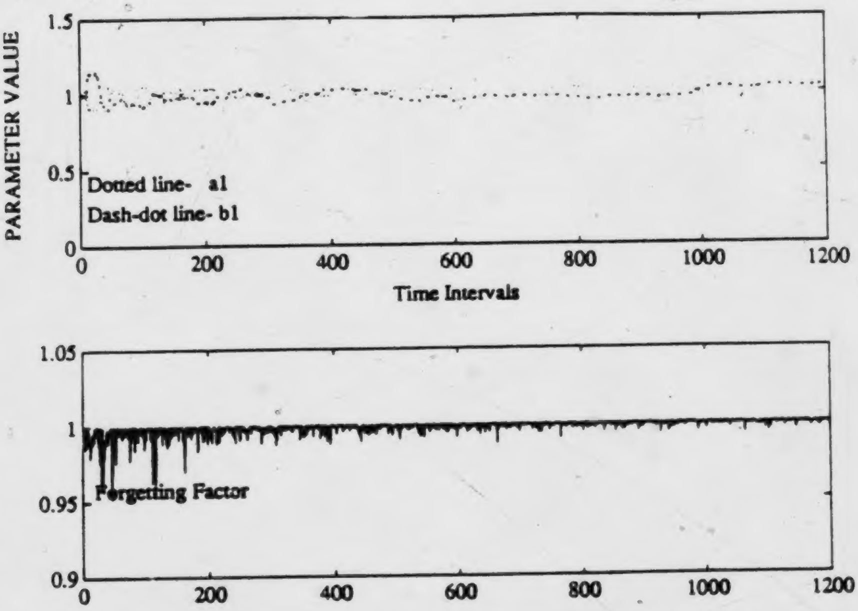

Figure 5.3c Parameter and Forgetting Factor Trajectories: Constant Determinant Forgetting Factor

Figures 5.4a,b and c illustrate the evolution of the covariance bound defined by equation (5.3), after each 100 intervals. Note that the initial ellipse is circular since $P(0) = I$, and the subsequent ellipses become more distorted in the direction perpendicular to the ϕ vector. The ellipses provide a clear understanding of how covariance windup occurs, as well as how it is affected by different choices of the forgetting factors. In the constant trace case the maximum dimension of the ellipse will be constrained while in the constant determinant case the area of the ellipses remains constant.

In the previous section it was concluded that if exponential forgetting is used under rank deficient conditions then covariance windup will necessarily result. In the case of constant trace and constant determinant forgetting factors, the growth becomes asymptotically slower as the matrix expands, because exponential forgetting ceases when the forgetting factor goes to unity. In the constant information case, the forgetting factor does not go to unity, therefore exponential forgetting is allowed to continue under rank deficient conditions, resulting in covariance windup.

It is concluded that the forgetting factors which were derived based on maintaining a matrix norm equal to a constant are more robust, since these cause the forgetting factor to go to unity under rank deficient conditions.

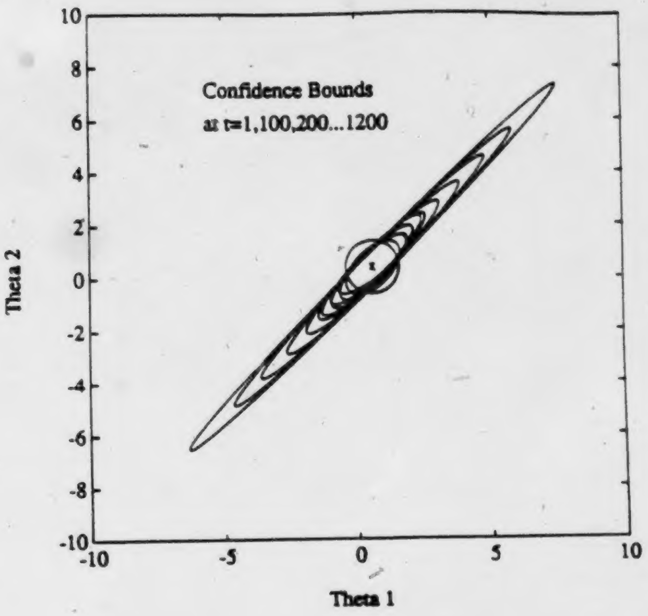

Figure 5.4a Covariance Bounds - Constant Information Forgetting Factor

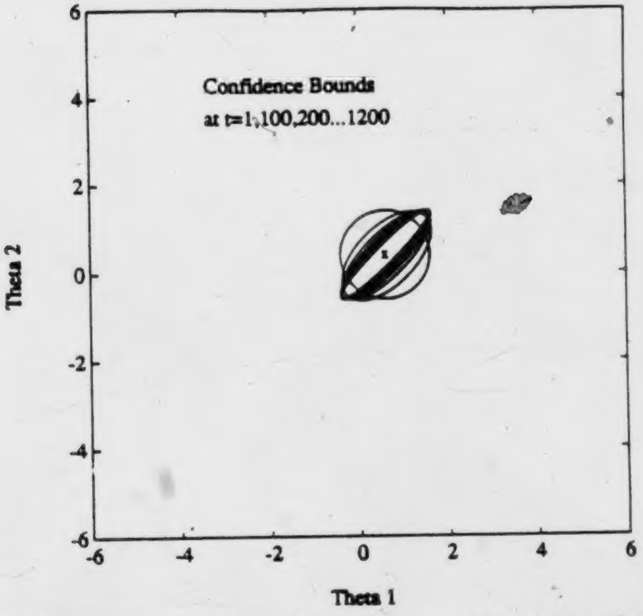

Figure 5.4b Covariance Bounds - Constant Trace Forgetting Factor

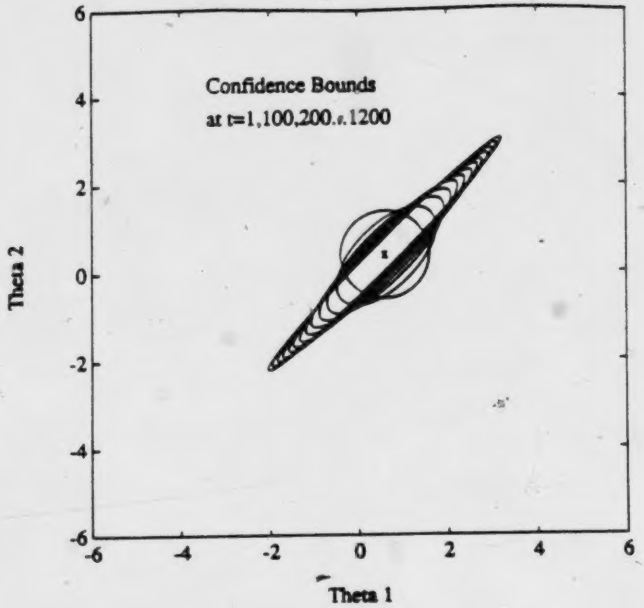

Figure 5.4c Covariance Bounds - Constant Determinant Forgetting Factor

Table 5.1 presents a summary of the determinant, trace and condition numbers of the P matrix after 400 intervals of the experiment. These results are provided as a numerical verification of the results.

Table 5.1 Summary of Final Conditions of the P Matrix

	Tr(P(400))	|P(400)|	Cond(P(400))
Initial Conditions	2.00	1.00	1.00
Constant Information	9.34	2.36	34.93
Constant Determinant	5.13	1.00	24.27
Constant Trace	2.00	0.25	14.01

Table 5.1 reveals that the constant trace forgetting factor takes the

most conservative approach, to using the information available to it since it maintains the smallest matrix trace of the three cases considered. The constant determinant method is apparently less conservative based on that measure. The constant information algorithm is apparently the least conservative by the previous arguments. In this case the trace of the matrix grows rapidly due to the covariance windup effect.

5.10 Experimental Evaluation

The three exponential forgetting factors previously described were applied to an off-line recursive estimation problem using a set of closed loop process data. The objectives were to verify the conclusions drawn from the simulation results by using operating data, and to demonstrate the practicality of the new constant determinant forgetting factor under more realistic conditions. The evaluations were all carried out using the same off-line data set in order to guarantee that other factors, such as random process disturbances, are held constant and do not influence the results.

The data were gathered from a distillation column separating a 50 mass percent methanol-water feed into a 95 mass percent methanol top product and a 5 mass percent bottom product. Figure 5.5 provides a schematic diagram of the equipment involved. The details of the column are provided elsewhere (Langman J. 1986, Pacey W.C. 1973, Svrcek W.Y. 1967). The column was interfaced to a LSI 11/23 microcomputer system which performed the composition control and the data collection functions.

All the comparisons presented in this evaluation, and those in

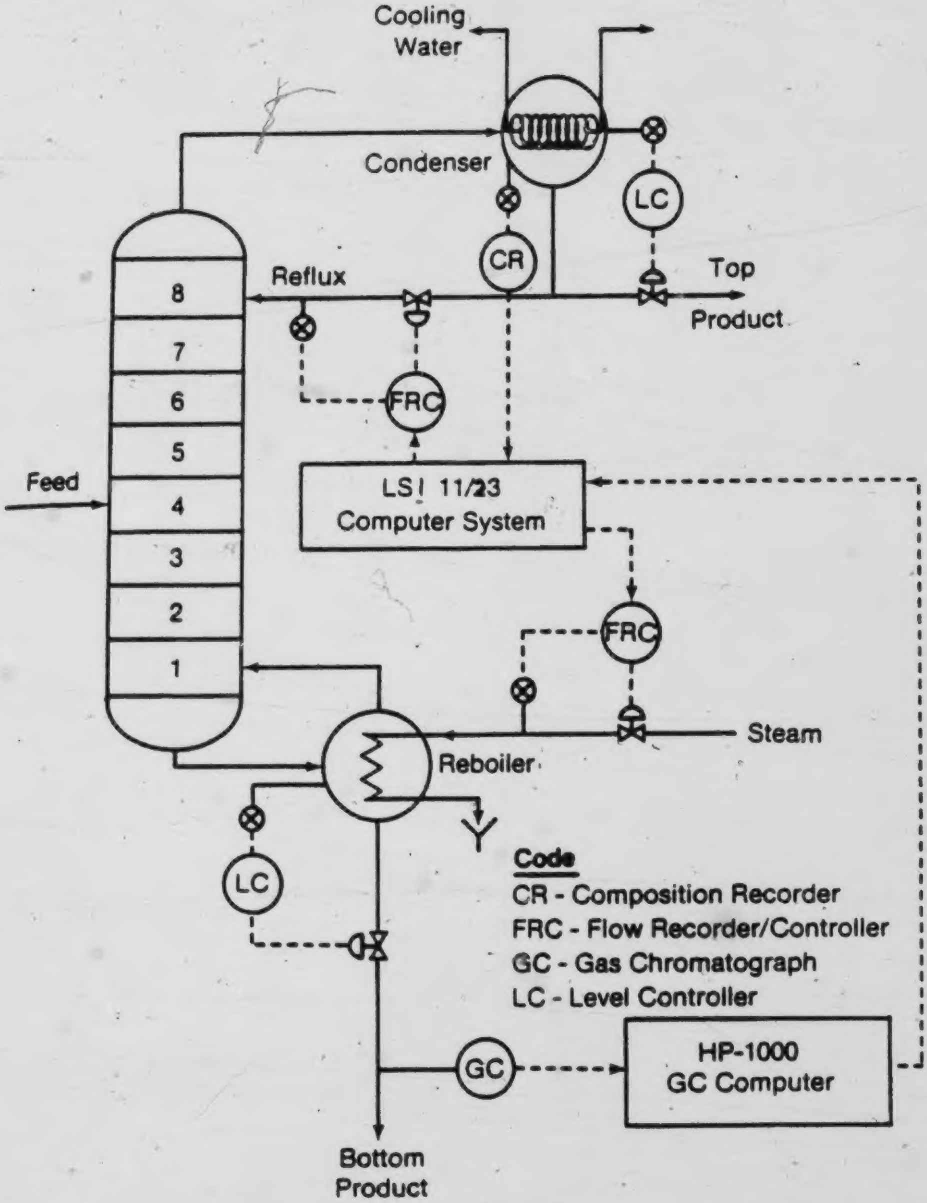
Cooling Water
Condenser
LC
CR
Reflux
Top Product
8
7
6
5
4
3
2
1
FRC
Feed
LSI 11/23 Computer System
FRC
Steam
Reboiler
LC
Code
CR - Composition Recorder
FRC - Flow Recorder/Controller
GC - Gas Chromatograph
LC - Level Controller
GC
HP-1000 GC Computer
Bottom Product

Figure 5.5 Schematic Diagram of the Pilot Scale Distillation Column

subsequent chapters use the same data set which was gathered from a test of a single-input, single-output, adaptive, proportional and integral controller of Vermeer (1987). During this test the distillate composition was controlled using the reflux flow. The bottom composition was regulated at 5 percent methanol by a fixed parameter PI controller, and the feed to the column was held constant.

The data, which is plotted in Figure 5.6, consists of 1000 measurements of the distillate composition and the reflux flow taken each minute. The setpoint of the overhead composition control was changed by ±1 percent every 150 minutes in a reversing step pattern.

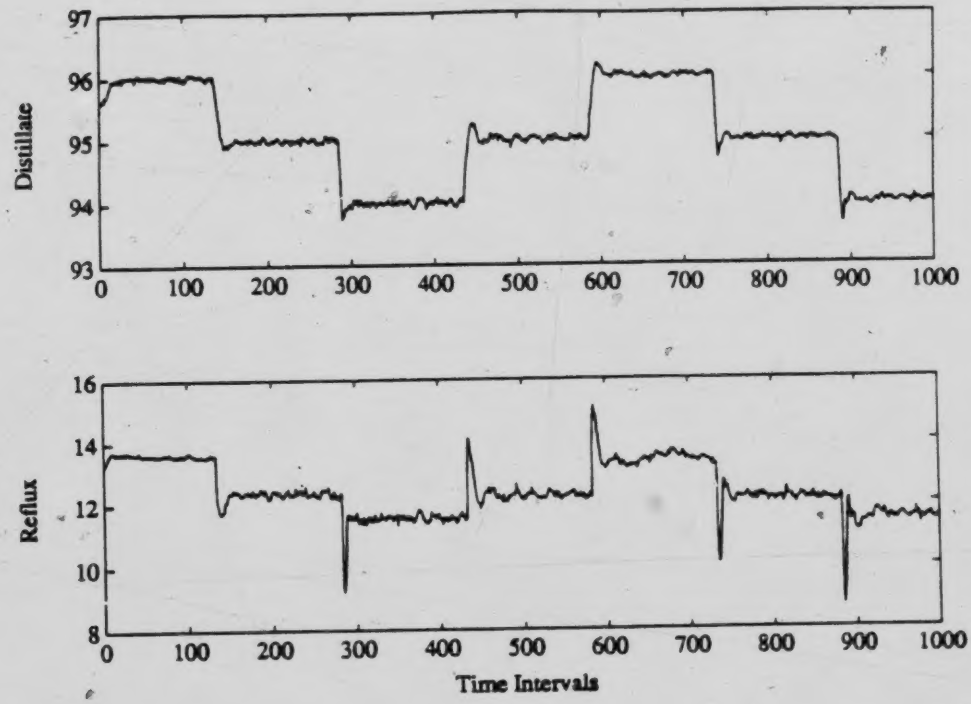

Figure 5.6 Distillation Column Data

because they were gathered under closed loop conditions and all the excitation is due to infrequent setpoint changes. Furthermore there is a significant level of noise on the measurement of the distillate composition which is due to incomplete mixing in the overhead accumulator.

The model of the system is assumed to be of the ARIMAX form, with first order A and C polynomials. The time delay was assumed to be one minute, and the order of the B polynomial was taken as zero. The model parameters a_1, b_0 and c_1 defined by equation (5.27) were estimated using the recursive ELS algorithm described in Chapter 4, with the three different types of forgetting factors described in this chapter.

$$\Delta y(t) = -a_1 \Delta y(t-1) + b_0 \Delta u(t-1) + (1 + c_1 q^{-1})\xi(t) \tag{5.27}$$

The data and parameter vectors were constructed as follows:

$$\phi(t) = \begin{bmatrix} -\Delta y(t-1) \\ \Delta u(t-1) \\ e(t-1|t-1) \end{bmatrix}, \quad \hat{\theta}(t) = \begin{bmatrix} \hat{a}_1 \\ \hat{b}_0 \\ \hat{c}_1 \end{bmatrix}$$

with the *a posteriori* prediction error calculated as:

$$e(t|t) = \Delta y(t) - \hat{\theta}(t|t)\phi(t)$$

Initial conditions for each test were chosen as follows:

$$P(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \hat{\theta}(0) = \begin{bmatrix} 0.99 \\ 0.01 \\ -0.8 \end{bmatrix}$$

The off-line evaluations of the recursive identification algorithms were conducted on an IBM PC-AT computer using the PC-Matlab package. An example of the actual code used is included in Appendix B.

The parameter estimates, forgetting factor, matrix determinant, and condition number for each run are plotted in Figures 5.7a,b,c,d, 5.8a,b,c,d, and 5.9a,b,c,d.

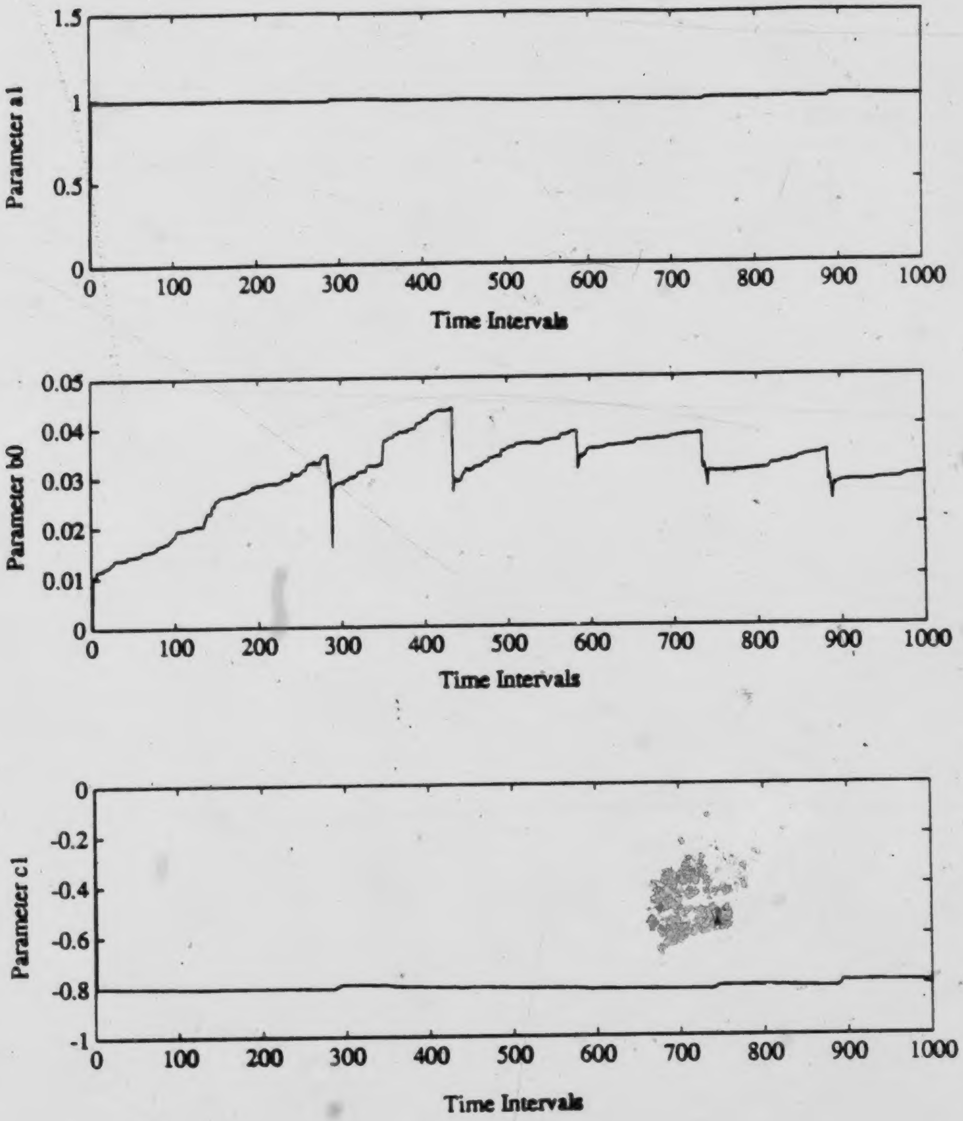

Figure 5.7a Parameter Trajectories: - Constant Information Forgetting Factor Applied to Distillation Column Data

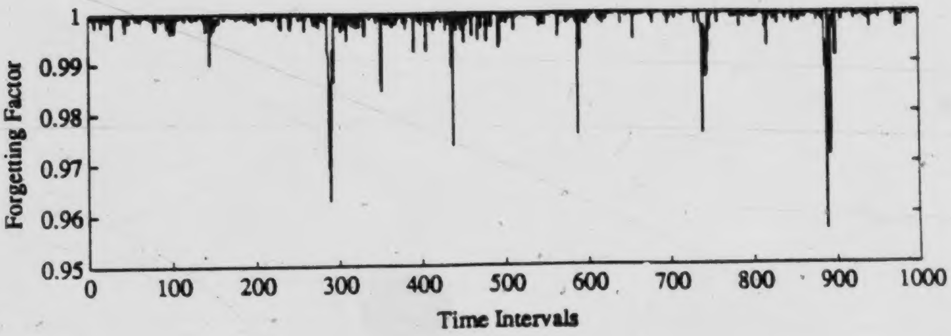

Figure 5.7b Forgetting Factor - Constant Information Forgetting Factor Applied to Distillation Column Data

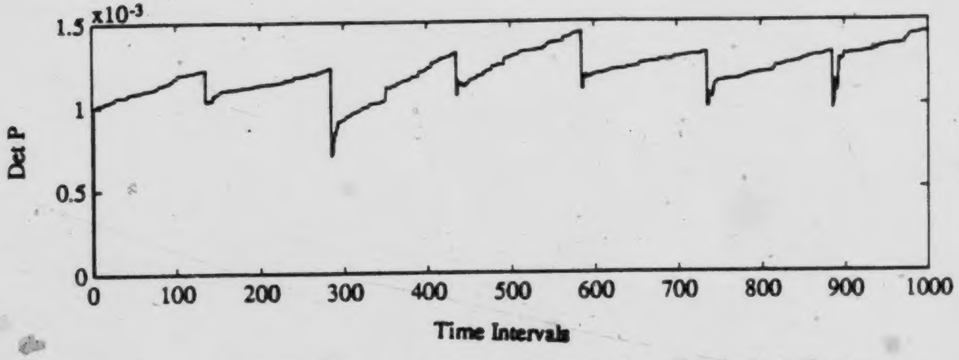

Figure 5.7c Matrix Determinant - Constant Information Forgetting Factor Applied to Distillation Column Data

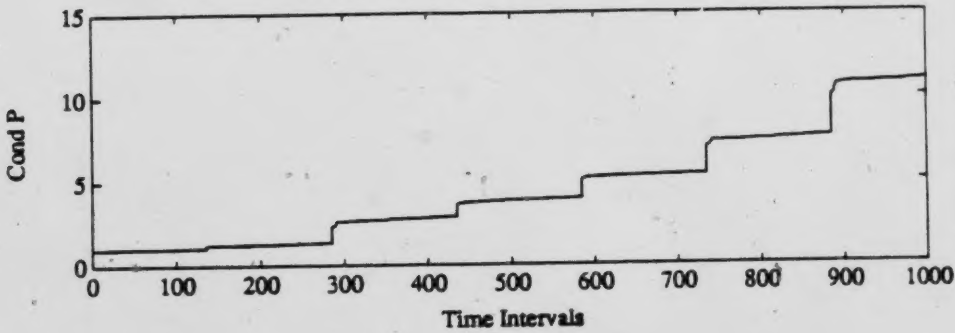

Figure 5.7d Condition Number - Constant Information Forgetting Factor Applied to Distillation Column Data

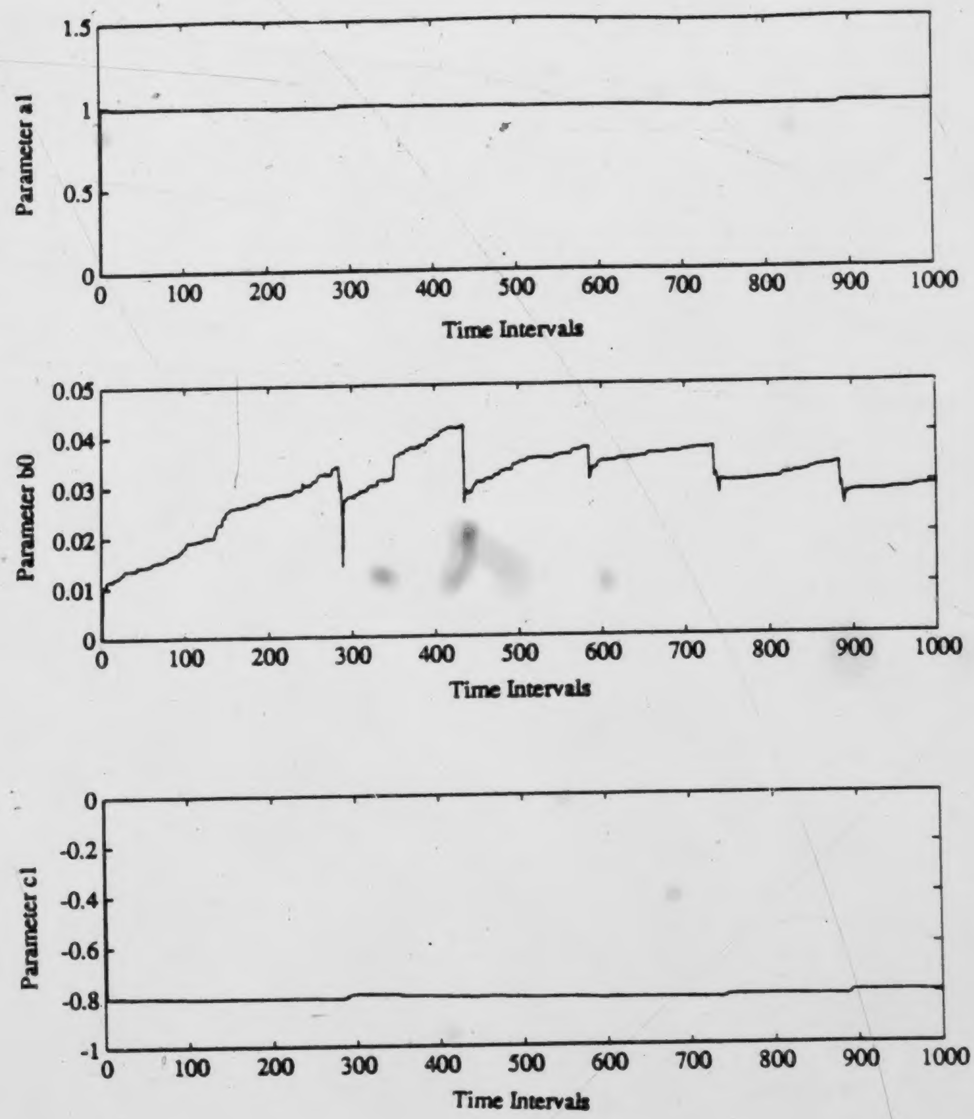

Figure 5.8a **Parameter Trajectories: Constant Trace Forgetting Factor Applied to Distillation Column Data**

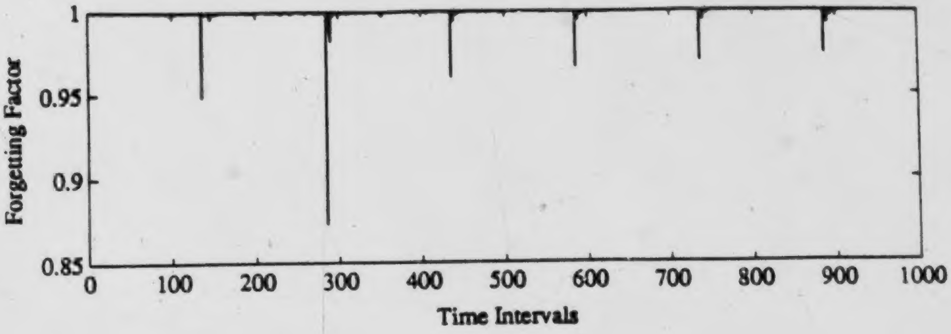

Figure 5.8b Forgetting Factor - Constant Trace Forgetting Factor Applied to Distillation Column Data

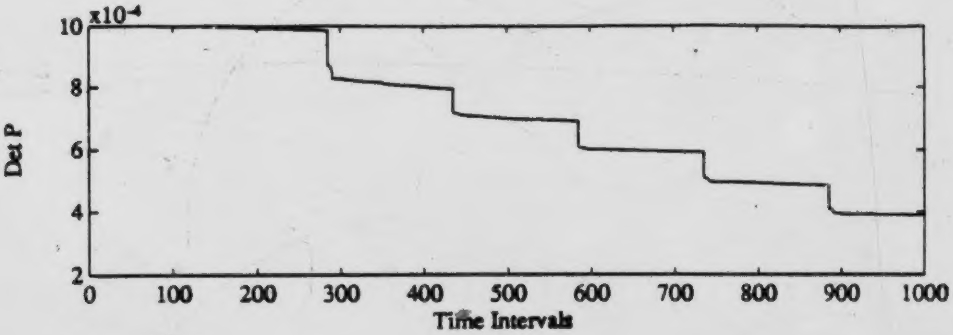

Figure 5.8c Matrix Determinant - Constant Trace Forgetting Factor Applied to Distillation Column Data

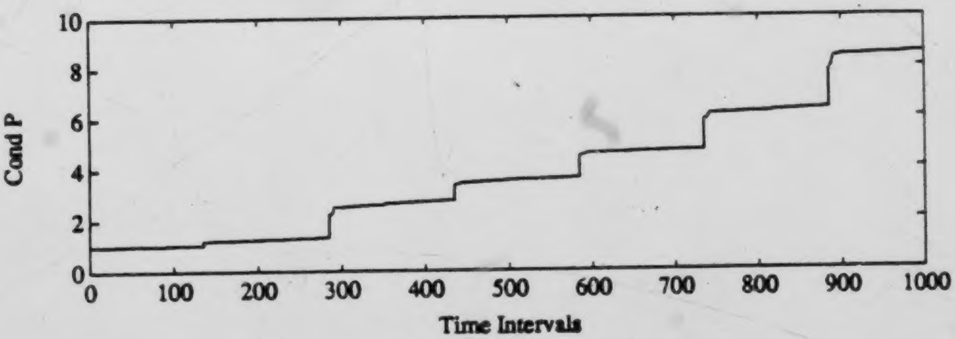

Figure 5.8d Condition Number - Constant Trace Forgetting Factor Applied to Distillation Column Data

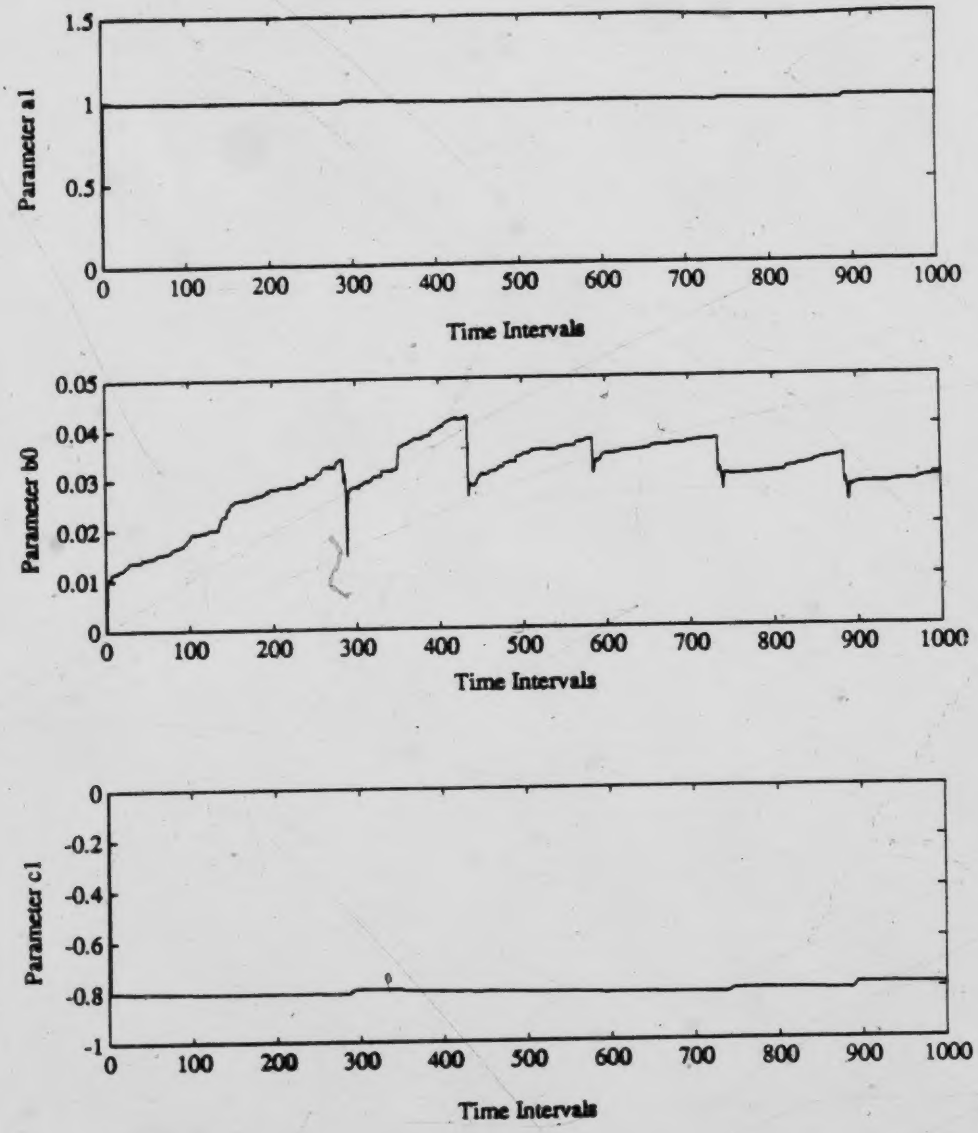

Figure 5.9a Parameter Trajectories: Constant Determinant Forgetting Factor Applied to Distillation Column Data

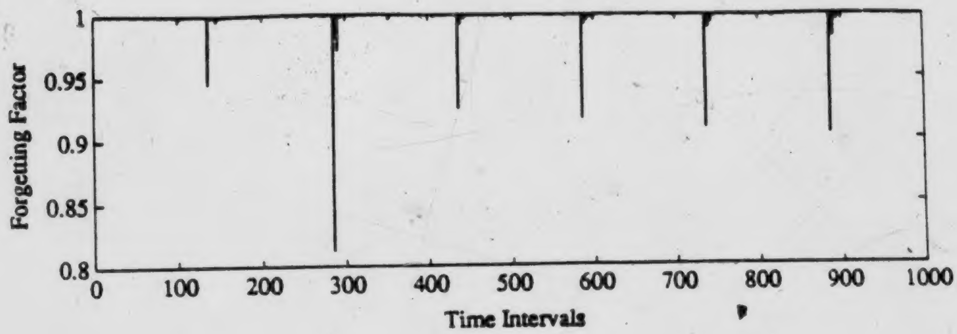

Figure 5.9b Forgetting Factor - Constant Determinant Forgetting Factor Applied to Distillation Column Data

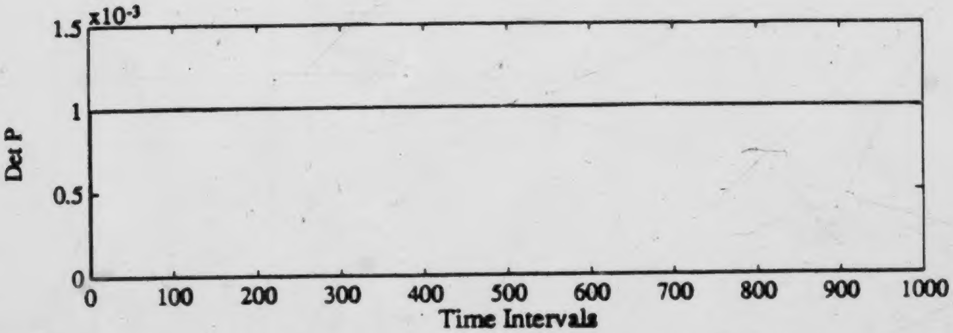

Figure 5.9c Matrix Determinant - Constant Determinant Forgetting Factor Applied to Distillation Column Data

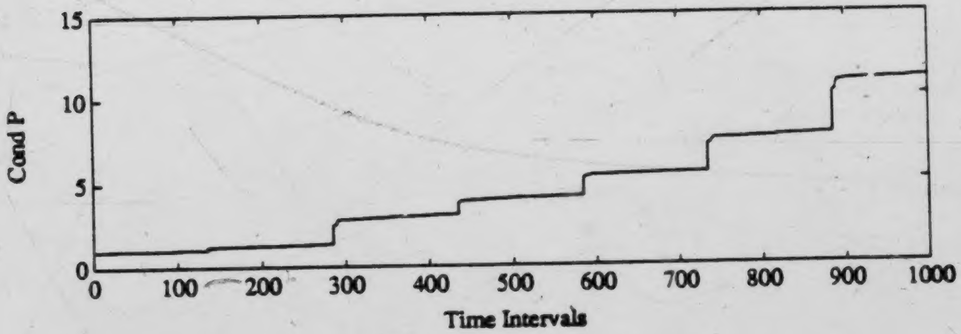

Figure 5.9d Condition Number - Constant Determinant Forgetting Factor Applied to Distillation Column Data

5.11 Experimental Observations

No significant difference was observed between the parameter estimates obtained using the three different forgetting factor methods. The differences are very subtle and may not have a noticeable affect on parameter estimates for thousands of intervals. The effects on the properties of the P matrix are, however, quite apparent and distinct in this evaluation. These effects are interesting because they reveal the impact of the forgetting factor on the properties of the P matrix under excited and non-excited conditions.

The forgetting factors calculated for the constant determinant and constant trace cases are very similar, remaining very close to unity during periods of no excitation, and decreasing to approximately 0.85 when setpoint changes occurred. In these cases the matrix properties (det{P}, Cond{P}) remain nearly constant when the process was quiet. In the constant information case, however, the forgetting factors were generally lower during the quiet periods because they were affected by process noise. As a result, the values of Tr{P} and |P(t)| tended to inflate, indicating covariance windup.

In all three cases the condition number increased dramatically during periods of rich excitation, but remained fairly constant when the process was quiet. When the process is quiet, the condition number is unaffected by exponential forgetting since under these conditions all the elements (hence the eigenvalues) of the P matrix are increased proportionately. For this reason, the condition number by itself does not provide a good indication of covariance windup.

5.12 Conclusions

Under both simulation and experimental conditions it was found that the constant information forgetting factor of Fortescue *et al.* (1981), the constant trace forgetting factor of Sripada and Fisher, (1987), and the new constant determinant forgetting factor were effective in maintaining estimator alertness. In the experimental example considered, the effect of different exponential forgetting factors on the parameter estimates was indistinguishable.

It was found that the constant trace and constant determinant forgetting factors tend to unity during periods of low excitation and are relatively unaffected by process noise, a result which was supported by the experimental as well as the simulation tests. It is concluded that these forgetting factor methods provide superior protection against the problem of covariance windup. In contrast, it was found that the constant information forgetting factor of Fortescue *et al.* (1981) is not effective in preventing covariance windup when the system is not excited, since process noise directly affects the forgetting factor.

5.13 Summary

Chapter 5 examines the use of exponential forgetting forgetting factors for maintaining estimator alertness. The problem of covariance windup and "bursting" are identified as potential hazards in the use of exponential forgetting under non-informative conditions.

Three different variable forgetting forgetting factors are presented and compared, each of which is derived by causing some measure of the P matrix to remain equal to a constant. The constant information approach of Fortescue *et al.* (1981) is found to be the least robust, since it does not cause forgetting to cease when the are data non-informative, and is sensitive to process noise. The constant trace algorithm of Sripada and Fisher (1987), was found to be a very robust and conservative approach which effectively prevents covariance windup. A new forgetting factor which is based on maintaining the determinant of the P matrix equal to a constant is derived and compared to the others. It is found that this approach is robust in the sense that forgetting is shut off when data are not informative, however it is a less conservative approach because it allows the trace of the matrix to increase.

The confidence bound ellipse is presented as a valuable tool for graphically examining a number of fundamental properties of the P matrix. The eigenvalues, eigenvectors, trace, determinant and condition number of a 2x2 matrix may be interpreted by examination of the ellipse.

A comparison between the three forgetting factor methods is made under simulated, closed loop proportional feedback conditions. These conditions provide a challenging test of the performance of the forgetting factor because they restrict the rank of the information matrix and the data are not persistently excited. The algorithms were compared based on the parameter trajectories and the behavior of the P matrix in time, graphically presented as ellipses. The results and conclusions drawn from the simulation were verified by application of the forgetting factors to a recursive identification problem using experimental data. It was found that the constant trace and constant determinant forgetting factors were robust and practical methods. The constant information forgetting factor however was considered less robust since it showed a tendency to allow covariance windup.

Chapter 6 examines an alternative idea to exponential data forgetting known as directional forgetting which, although simple, is an elegant approach to the problem. Comparisons are made to the results presented in this chapter.

Chapter 6

Directional Forgetting

6.1 Introduction

In Chapter 5 the idea of exponential forgetting was introduced which applied a factor $\lambda(t)$ to previous data which increases the elements of the P matrix in a uniform manner. A novel approach to maintaining alertness, derived from Bayesian arguments applied to parameter estimation is directional forgetting (Peterka, 1981; Kulhavy and Karny, 1984), (also termed restricted exponential forgetting (Kulhavy, 1987)). The directional forgetting idea is based on the assertion that only the "evidently" obsolete piece of posterior knowledge should be updated, hence the P matrix should only be modified in the direction of the current Kalman gain vector $P(t-1)\phi(t)$. The resulting algorithm works in such a manner that only those directional components of the P matrix which are improved or innovated by the vector $\phi(t)$ are taken as obsolete and therefore replaced. The main advantage of this approach over exponential forgetting is that it avoids the restriction that the system be uniformly excited. Those directions which are unexcited are unaffected by forgetting, therefore the covariance windup problem is inherently avoided.

Chapter 6 provides a brief discussion of the motivating theory behind directional forgetting. The rigorous derivation of the algorithm is necessarily complicated and is provided elsewhere (Kulhavy, 1987). The resulting algorithm is interpreted and compared to exponential forgetting.

The algorithm was applied to the simulation example of Chapter 5 to allow a direct comparison to the exponential forgetting results. Application to the distillation column data are presented as a further verification, and a demonstration of the practicality of the method under realistic conditions.

6.2 The Bayesian Approach to Identification

The conditional probability density function (CPDF) of an event y(t) conditioned on the observation of a data value at time t denoted by $\phi(t)$, and a set of previously gathered observations D, is denoted by:

$$\mathcal{P}_{t/t-1}(y(t)|\phi(t),D) \tag{6.1}$$

If u(t) is the only observed input to a process then the input-output relation:

$$y(t) = \theta^T\phi(t) + \xi(t) \tag{6.2}$$

is written in terms of the CPDF of y(t) as:

$$\mathcal{P}_{t/t-1}(y(t)|u(t),\theta) \tag{6.3}$$

If the input u(t) is independent of the system parameters (Peterka, 1981) such that:

$$\mathcal{P}_{t/t-1}(u|\theta) = \mathcal{P}_{t/t-1}(u) \tag{6.4}$$

then Bayes rule may be applied to system identification, resulting in an expression for the CPDF of the parameter estimate $\hat{\theta}(t)$ conditioned on observed data given by:

$$\mathcal{P}_{t/t}(\theta) \propto \mathcal{P}_{t/t-1}(y|u,\theta)\cdot\mathcal{P}_{t/t-1}(\theta) \qquad t=1,2\ldots \tag{6.5}$$

If the parameters are considered to be time varying then evaluation of the CPDF $\mathcal{P}_{t/t}(\theta)\rightarrow\mathcal{P}_{t+1/t}(\theta)$ must be computed to complete the recursion. An exact evaluation of this requires a probabilistic model of the time variation of the parameters. Ordinarily, only a rough description of the behavior of the parameters is available, e.g.: "the parameters vary slowly", and an heuristic procedure must be used. The simplest way to consider that the parameters change in the next interval is to directly increase the uncertainty of old estimates $\mathcal{P}_{t/t}(\theta)$ by the "flattening" operation (Peterka, 1981):

$$\mathcal{P}_{t+1/t}(\theta) \propto \left[\mathcal{P}_{t/t}(\theta)\right]^{\lambda} \tag{6.6}$$

where λ is a forgetting factor which may be time dependent. The directional forgetting idea, however observes that there are some directions in which the CPDF of unknown parameters remains constant (due to the lack of new information) and should not be arbitrarily increased. The exponential forgetting factor reduces the whole CPDF, $\mathcal{P}_{t/t}(\theta)$, and as a result the algorithm can become very sensitive to even small changes in the information content of the data, leading to covariance windup and possible "blowup" as described in Section 5.2.

The directional forgetting algorithm (Kulhavy and Karny, 1984) determines a parameter subspace on which the probability distribution remains unchanged by the recent data. This region is spanned by the non-excited directions or modes of the process. Ordinary exponential forgetting is applied on the complimentary parameter subspace. The two regions are chosen to be stochastically independent so that "forgetting" in one subspace does not affect the other.

The development of the directional forgetting algorithm considers the existence of an operator $\mathcal{F}_t$ (dependent on the direction of the data) which maps the parameter space into two conditionally independent subspaces, i.e. the generic point θ in parameter space is transformed into Φ such that:

$$\Phi = \left\{\Phi_1, \Phi_2\right\} = \mathcal{F}_t(\theta) \tag{6.7}$$

The conditional independence of the two CPDFs is expressed by:

$$\mathcal{P}_{t/t}(\Phi) = \mathcal{P}_{t/t}(\Phi_1)\cdot\mathcal{P}_{t/t}(\Phi_2) \tag{6.8}$$

and

$$\mathcal{P}_{t/t-1}(\Phi) = \mathcal{P}_{t/t-1}(\Phi_1)\cdot\mathcal{P}_{t/t-1}(\Phi_2) \tag{6.9}$$

The recent data $\phi(t)$ does not affect the CPDF of Φ_1, therefore :

$$\mathcal{P}_{t+1/t}(\Phi_1) = \mathcal{P}_{t/t}(\Phi_1) = \mathcal{P}_{t/t-1}(\Phi_1) \tag{6.10}$$

Ordinary exponential forgetting is applied to the complimentary subspace Φ_2 by:

$$\mathcal{P}_{t+1/t}(\Phi_2) \propto \mathcal{P}_{t/t}(\Phi_2) = \left[\mathcal{P}_{t/t-1}(\Phi_2)\right]^{\lambda} \tag{6.11}$$

The directional forgetting scheme updates $\mathcal{P}_{t/t-1}(\Phi)$ as:

$$\mathcal{P}_{t+1/t}(\Phi) = \mathcal{P}_{t+1/t}(\Phi_1)\cdot\mathcal{P}_{t+1/t}(\Phi_2)$$

$$\propto \mathcal{P}_{t/t-1}(\Phi_1)\cdot\left[\mathcal{P}_{t/t}(\Phi_2)\right]^{\lambda} \tag{6.12}$$

The derivation of a practical form of the directional forgetting algorithm from this formulation is necessarily difficult and is described in the paper by Kulhavy (1987). The algorithm, as it applies to least squares is however, quite simple, and is presented in the following section.

6.3 Directional Forgetting Applied to the Least Squares Method

The matrix update equations (Kulhavy and Karny, 1984) which correspond to the update of the CPDF, equations (6.10), (6.11), and (6.12) are:

$$P^{-1}(t|t) = P^{-1}(t|t-1) + \phi(t)\phi^T(t) \tag{6.13}$$

$$P^{-1}(t+1|t) = P^{-1}(t|t) + (\alpha(t) - 1)\phi(t)\phi^T(t) \tag{6.14}$$

with parameter estimates given by:

$$\theta(t+1|t) = \hat{\theta}(t|t-1) + \frac{P(t|t-1)\,\phi(t)\epsilon(t)}{1 + G(t)} \tag{6.15}$$

In practice, the update and forgetting steps given by equations (6.13) and

(6.14) may be combined. Kulhavy (1987) also introduces an updating criterion δ_2 which prevents division by zero when the value of α becomes very small. Applying the matrix inversion lemma to the resulting algorithm gives:

if $|\alpha(t)|>\delta_2$:

$$P(t+1|t) = P(t|t-1) - \frac{P(t|t-1)\phi(t)\phi^T(t)P(t|t-1)}{1/\alpha(t) + \phi^T(t)P(t|t-1)\phi(t)} \tag{6.16a}$$

if $|\alpha(t)|\leq\delta_2$:

$$P(t+1|t) = P(t|t-1) \tag{6.16b}$$

The weighting factor $\alpha(t)$ is computed as a function of the forgetting factor $\lambda(t)\in(0,1]$ and $G(t)$ as:

$$\alpha(t) = \lambda(t) - \frac{1 - \lambda(t)}{G(t)} \tag{6.17}$$

This choice ensures that the matrix $P(t+1|t)$ remains positive definite for any $G(t)>0$ as long as the condition $\lambda\in(0,1]$ holds (see Appendix C for a proof of this). It is also of interest to note that the value of $\alpha(t)$ will always be less than $\lambda(t)$.

In equation (6.16) the update of $P(t|t-1)$ is suppressed when $|\alpha(t)| \leq \delta_2$, where $0 < \delta_2 << 1$, because in that case the effects of data updating $P(t-1|t)\rightarrow P(t|t)$ and directional forgetting $P(t|t)\rightarrow P(t+1|t)$ are exactly offset by one another. In practice it is recommended that the value of δ_2 be chosen in the interval $(10^{-2},10^{-6})$ depending on machine precision (Kulhavy and Karny, 1984).

If the value of $G(t)$ is zero then the data are "singular", in the sense that they contain no useful information and the updating/forgetting steps should cease entirely. It is necessary to introduce an on/off criterion to ensure that such singular data points are discarded, to avoid division by zero in equations (6.16) and (6.17). If $G(t) < \delta_1$ then equations (6.15) and (6.16a,b) are replaced by:

$$\hat{\theta}(t) = \hat{\theta}(t-1) \tag{6.18}$$

$$P(t+1|t) = P(t|t) = P(t|t-1) \tag{6.19}$$

6.4 Choice of the Forgetting Factor

A critical identifiability requirement in any problem is that the degrees of freedom defined by N-n is greater than 1, or that the number of data observations be greater than or equal to the number of parameters. In recursive problems, the number of data observations is usually very large, and the forgetting factor defines a window of recent data observations whose length is approximated by the mean sample length (MSL) defined as $\lambda/(1-\lambda)$ (Ljung and Soderstrom, 1973). Exponential forgetting is applied to the whole parameter space hence the value of the MSL must be related to the total number of parameters. Directional forgetting, however, applies the usual exponential type forgetting to only a rank one subspace of the P matrix, which corresponds to identification of only one parameter, therefore the MSL may be chosen on a one parameter basis and the value of the forgetting factor is independent of the number of parameters in the problem. For instance if a forgetting factor of 0.95 is used for a three parameter problem using exponential forgetting, then the mean sample length is 19. Assuming that the data uniformly span the parameter space, then the corresponding directional forgetting factor depends on the number of samples per parameter which in this case is 6.33 and the directional forgetting factor is 0.86. Since the forgetting factor relates directly to the MSL of each individual parameter in the system, it is a more meaningful tuning knob than those proposed for exponential forgetting in Chapter 5.

Kulhavy and Karny (1984) observed that the directional forgetting algorithm is less sensitive to the choice of forgetting factor than standard exponential forgetting factors due to the inherent robustness of the formulation. It was found however that periodic adjustment of this factor was beneficial to speed convergence. Kulhavy (1987) proposed a time varying forgetting factor λ(t) which is derived from a mathematical formulation of the time updating model given by equation (6.6). This uses the prediction error to lower the forgetting factor when prediction errors are large, and correspondingly raise the forgetting factor when they are small.

$$\lambda(t) = \left[1 + (1+\rho)\left[\ln(1+G(t)) + \frac{G(t)(\epsilon_N^2(t)-1)}{1+G(t)}\right]\right]^{-1} \tag{6.20}$$

The normalised square of the prediction error ϵ_N^2 is given by:

$$\epsilon_N^2(t) = \frac{\epsilon^2(t)}{\sigma^2(1+G(t))} \tag{6.21}$$

The value ρ is a tuning knob which is roughly related to a "steady state" forgetting factor λ_{ss} by the relation:

$$\lambda_{ss} \approx \left[1 + 2\rho \right]^{-1} \tag{6.22}$$

The value of λ_{ss} is derived from (6.20) by considering that under true steady state conditions the covariance update becomes insignificant:

$$\phi^T(t)P(t|t-1)\phi(t) \cong \phi^T(t)P(t+1|t)\phi(t) \tag{6.23}$$

and the normalised prediction error should approach unity:

$$\epsilon_N^2 \cong 1 \tag{6.24}$$

Equation (6.22) is a relation which makes it possible to choose a value for ρ by means of the value of λ_{ss} which may be easier to specify. Based on a series of simulated experiments with an adaptive controller, Kulhavy (1987) concluded that a value of $\rho \cong 0.2$ resulted in good performance over a wide range of conditions. This value corresponds to $\lambda_{ss} \cong 0.714$, or a MSL of 2.5, which is significantly lower than the values normally used for exponential forgetting factors (0.95-1.0).

Kulhavy (1987) notes that the forgetting factor given by equations (6.20) and (6.21) is comparable in some respects to that of Fortescue *et al.* (1981) since the result of the later may be derived from equation (6.20) by certain simplifications and approximations. The constant information idea of Fortescue *et al.* (1981) applied to equations (5.16),(5.17) and (5.18) results in the forgetting factor (Kulhavy, 1987):

$$\lambda(t) = \left[1 + \frac{1}{\Psi_0}\epsilon_N^2(t)\right]^{-1} \qquad (6.25)$$

where the factor Ψ_0 is chosen *a priori*. The similarity to equation (6.20) is revealed by considering Ψ_0 to be time varying, given by $N_0=(1 + 1/G(t))/(1-\rho)$. The resulting formula is an approximation of (6.20):

$$\lambda(t) = \left[1 + \frac{(1-\rho)G(t)\epsilon_N^2}{(1 + G(t))}\right]^{-1} \qquad (6.26)$$

In an experimental comparison (Kulhavy, 1987) concluded that performance of the directional forgetting algorithm was superior to the constant information approach of Fortescue *et al.* (1981) and in general much less sensitive to the heuristic factor ρ than the corresponding factor $1/\Psi_0$.

6.5 Interpretations of the Directional Forgetting Method

The directional forgetting algorithm described by equations (6.15), (6.16) and (6.17) has distinct advantages over exponential forgetting since its forgetting factor does not affect the entire P matrix as it does in the exponential forgetting algorithms. The algorithm differs markedly, however from the usual form of weighted least squares algorithm, since α is allowed to take negative values, hence forget data, when G(t) is small. Its effect is best understood by considering the P matrix in terms of the confidence bound ellipsoid, and examining the effect of the data update and the forgetting steps separately. Figure 6.1 provides such an illustration for a two ARMA parameter system. The data update step given by equation (6.13) causes the diameter of the ellipsoid to be reduced in the direction of the Kalman gain vector $P(t|t-1)\phi(t)$. The forgetting step, corresponding to equation (6.14), will involve subtraction from the $P^{-1}(t|t)$ matrix, since $(\alpha(t)-1)$ is always negative, therefore the diameter of the covariance ellipsoid will be increased in the direction of the vector $P(t|t-1)\phi(t)$.

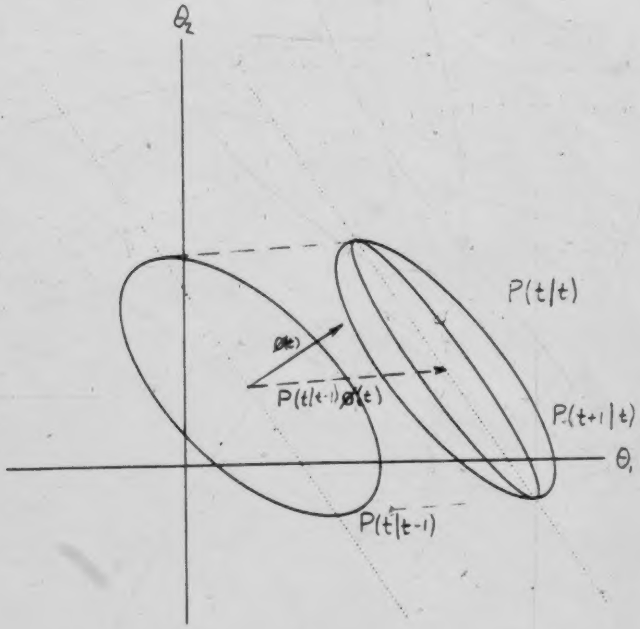

Figure 6.1 Effect of Directional Forgetting on the Confidence Bound Ellipse

The forgetting step in the directional forgetting algorithm is "data dependent" since the value of the data weight $\alpha(t)$ is calculated based on the information measure $G(t)$. The following three cases illustrate how the forgetting factor acts as a "tuning knob" which distinguishes useful data from redundant data, based on the value of $G(t)$.

1. At the point where $1/G(t) = \lambda/(1-\lambda) =$ MSL the data is redundant, the weighting $\alpha(t)$ is zero, and no covariance update takes place.

2. When $G(t) > (1-\lambda)/\lambda$ the data vector is relatively large, the value of $\alpha(t)$ will be positive, and the covariance matrix P is correspondingly shrunk in the direction of $P(t|t-1)\phi(t)$.

3. When $G(t) < (1-\lambda)/\lambda$ the data is "small" and $\alpha(t) < 0$, hence the covariance matrix is correspondingly increased in the direction of $P(t|t-1)\phi(t)$.

6.6 Reduced Rank Data

An important difference between the directional and exponential forgetting algorithms is the requirement of "persistence of excitation" to prevent covariance windup. In Chapter 5 it was demonstrated that exponential forgetting requires that the data be persistently exciting in order to cause the P matrix to converge to a constant non-singular matrix. It is easily shown that this restriction does not apply to directional forgetting. Recall the example from Chapter 5 in which the the data vectors from a system over a period of time $i=0 \rightarrow k$ do not span the entire parameter space, but are orthogonal to a vector x. The matrix update equation is written in terms of P^{-1} as:

$$P^{-1}(t) = P^{-1}(0) + \sum_{i=0}^{k} \alpha(i)\phi(i)\phi^{T}(i) \qquad (6.27)$$

Pre-multiplying both sides of equation (6.27) by the vector x^T, and post-multiplying by x gives:

$$x^T P^{-1}(t)x = x^T P^{-1}(0)x + x^T \left[\sum_{i=0}^{k} \alpha(i)\phi(i)\phi^T(i) \right] x \tag{6.28}$$
$$= x^T P^{-1}(0)x$$

It is clear that the components of the P^{-1} matrix which correspond to the null space of the data matrix will remain constant. This is seen by noting that the second term on the right hand side of (6.28) is equal to zero, hence the left side will remain constant, equal to the initial value, $x^T P^{-1}(0)x$. From this result it is straightforward to demonstrate that the value of $x^T P(t)x$ will never reach zero. This result is totally independent of the choice of the weighting factors $\alpha(t)$, hence it is also a property of the ordinary recursive least squares method. From this result it can be seen that the variable forgetting factor scheme of Kulhavy (1987), given by equations (6.20) and (6.21) will not cause the covariance windup problems, as did that of Fortescue *et al.* (1981), even though it introduces a prediction error term into the equations.

6.7 Simulation Example

The directional forgetting algorithm defined by equations (6.15), (6.16) and (6.17) was applied to the simulated closed-loop proportional feedback example of Chapter 5 to allow direct comparison to the exponential forgetting algorithms. The value of the forgetting factor was calculated by equation (6.20), with ρ=0.2. Figure 6.2a,b,c contain the

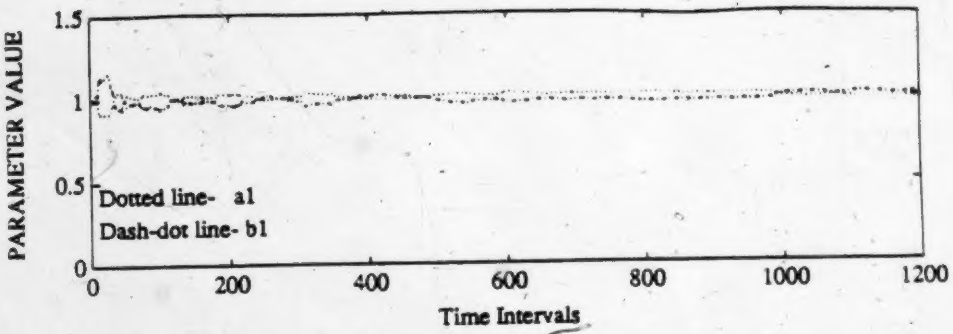

Figure 6.2a
Parameter trajectories for the Directional Forgetting algorithm applied to the simulation example.

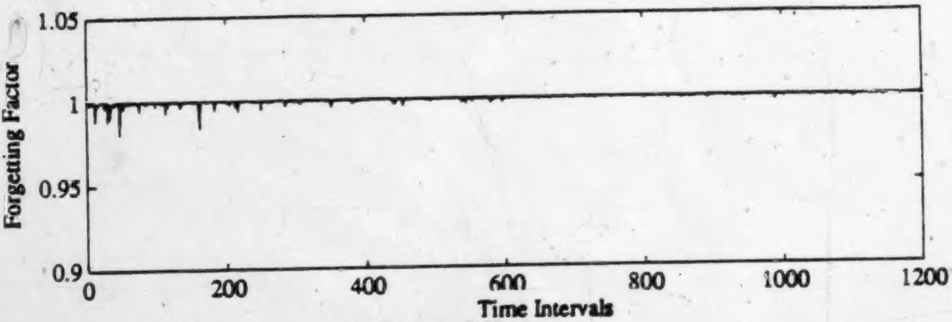

Figure 6.2b
Forgetting factor trajectory for the Directional Forgetting algorithm applied to the simulation example.

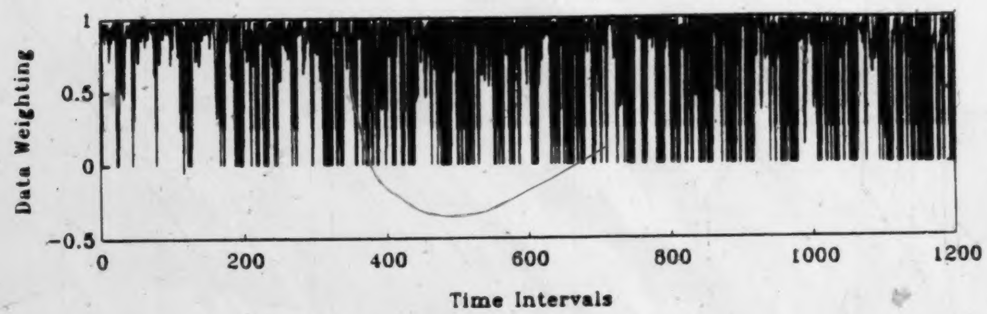

Figure 6.2c
Data weighting factor trajectory for the DF algorithm applied to simulation example.

parameter and forgetting factor trajectories, as well as a plot of the weighting factor $\alpha(t)$. Figure 6.3 illustrates the evolution of the this example is that one diameter of the covariance ellipse remains unaffected during the course of the experiment, because the data vectors provide no information about that direction. In each of the exponential forgetting cases this direction was observed to inflate to some extent. It is also interesting to note that in Figure 6.2c the data weighting factors approach zero over the course of the simulation. The data available from the system becomes redundant because the excitation is restricted by a simple proportional feedback controller.

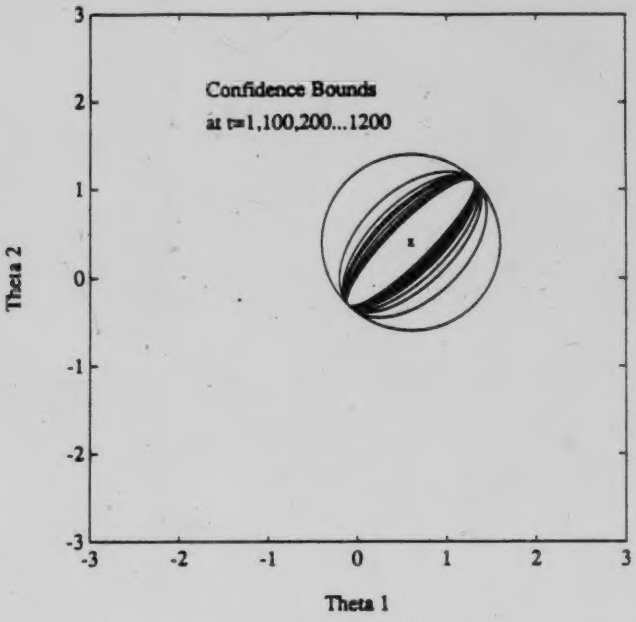

Figure 6.3
Evolution of the Covariance Bounds - Directional Forgetting applied to the simulation example.

6.8 Experimental Evaluation

The directional forgetting algorithm was applied to the same off-line distillation column data set as was used in Section 5.11, in order to directly compare its performance to exponential forgetting, and to demonstrate the practicality of the algorithm under realistic conditions.

The choice of model structure and initial conditions were identical to those used for the examples in Chapter 5. The directional forgetting algorithm defined by equations (6.15), (6.16) and (6.17) was used, with the variable forgetting factor calculated by equation (6.20), with $\rho = 0.2$.

The parameter trajectories, forgetting factors and data weighting factors of the experiment are presented in Figure 6.4a,b, and c. The values of $|P|$ and cond(P) are plotted in Figures 6.4 d and e.

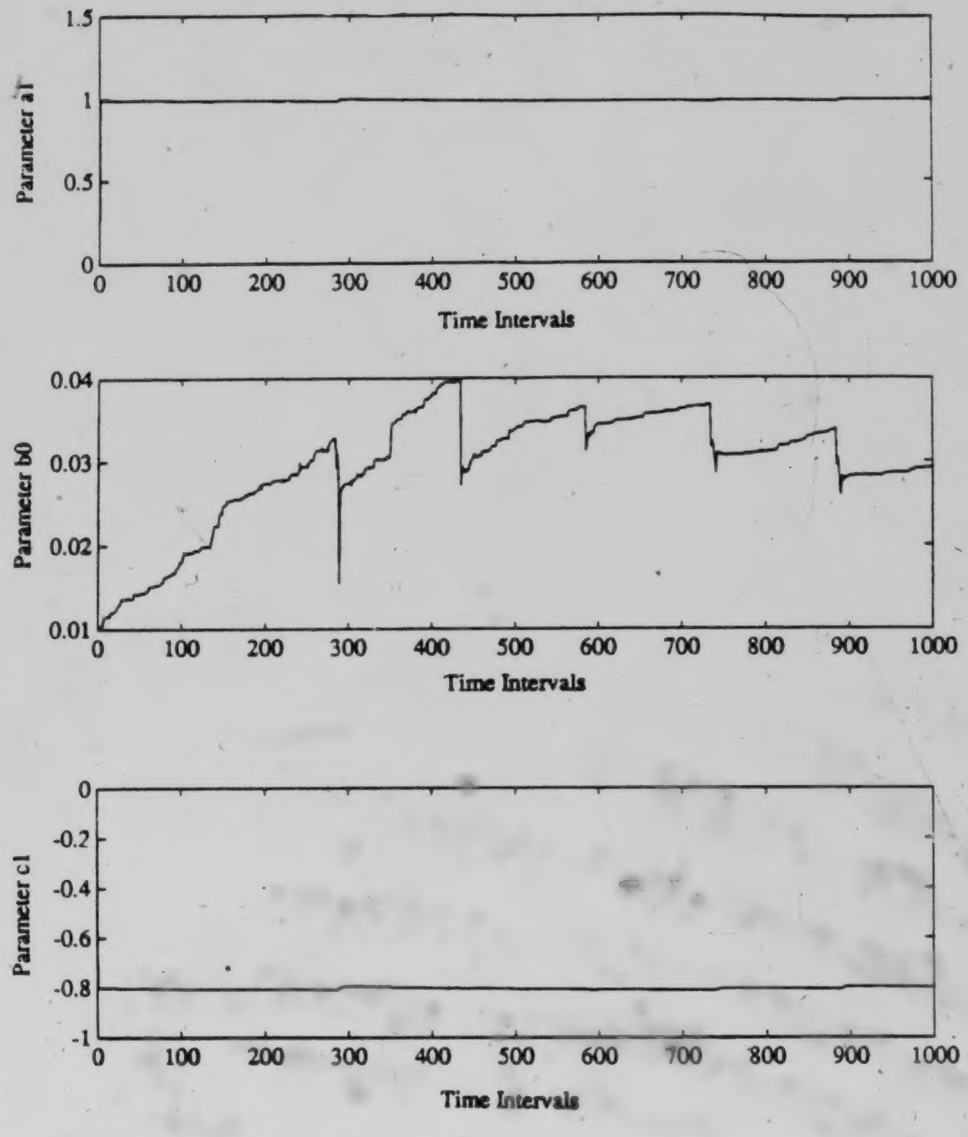

Figure 6.4 a
Parameter trajectories- Directional Forgetting applied to distillation column data.

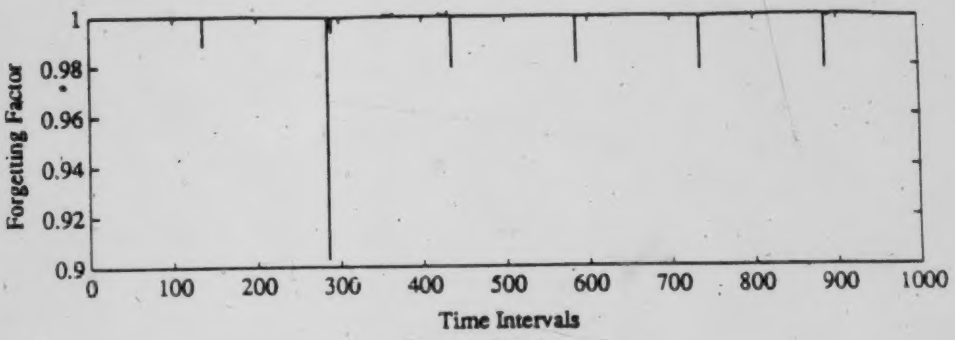

Figure 6.4 b

Forgetting factors - Directional forgetting applied to distillation column data.

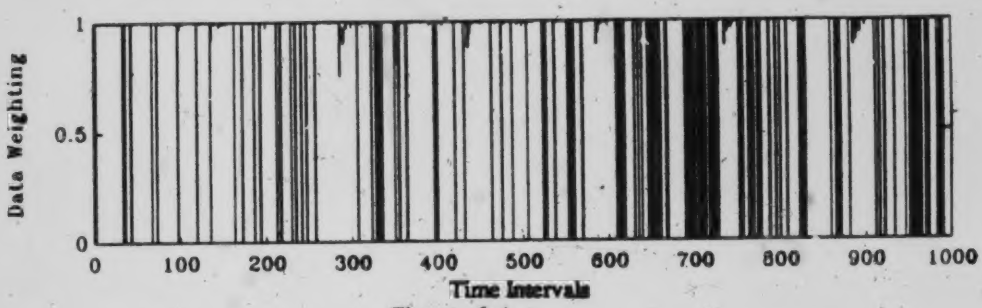

Figure 6.4 c

Weighting Factors- Directional forgetting applied to distillation column data.

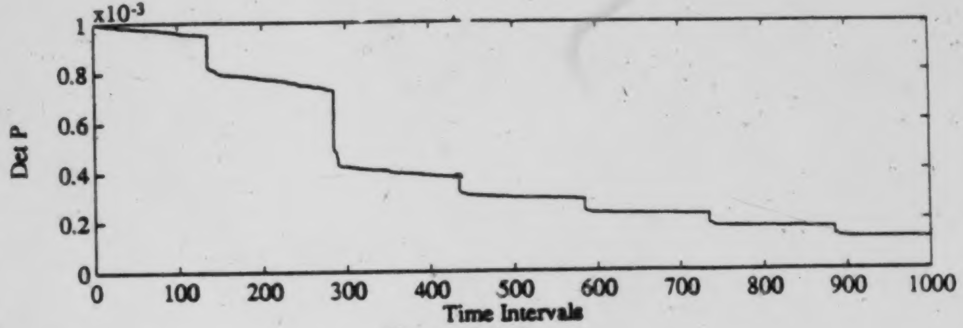

Figure 6.4 d

|P(t)| - Directional forgetting applied to distillation column data.

5

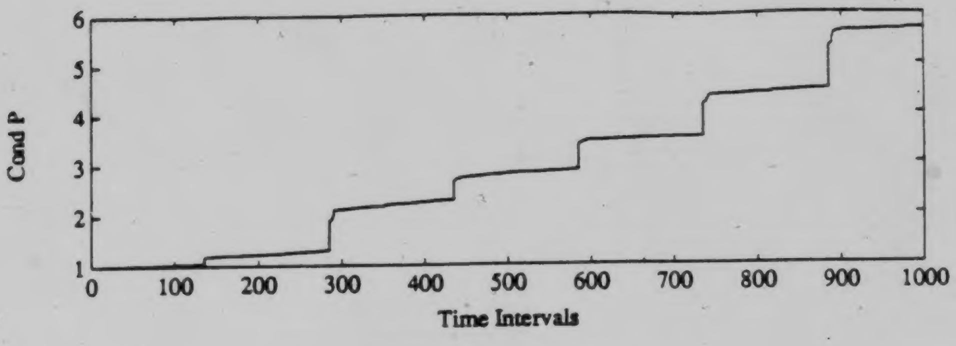

Figure 6.4 e

Cond{P(t)} - Directional forgetting applied to distillation column data.

The parameter trajectories exhibit no apparent differences from those in the exponential forgetting examples.

The forgetting factor remains very close to unity between setpoint changes, indicating that no forgetting takes place when the system is not excited. The plots of the matrix properties indicate no indication of covariance windup.

6.9 Summary

The directional forgetting algorithm (Kulhavy and Karny, 1984) is built on the assertion that old data should not be forgotten or reduced in importance until there is new information with which to replace it. Such an approach offers fundamental advantages over the ordinary exponential forgetting algorithm presented in Chapter 5, because the problem of covariance windup is inherently avoided by the structure of the resulting equations. An illustrative example is presented which demonstrates that the directional forgetting method is robust to rank deficient data.

It is observed that the value of the forgetting factor in the directional forgetting formulation is not related to the number of parameters in the identification problem, because it applies to only a rank one subspace of the P matrix. This is in contrast to the ordinary exponential forgetting factor which, in general increases to the power of n, the dimension of the problem.

A variable forgetting factor (Kulhavy, 1987) specific to the directional forgetting algorithm has been developed to adjust the rate of adaptation based on the current value of the normalised prediction error.

It is concluded that directional forgetting offers at least two distinct advantages over exponential forgetting:

1. The method is not subject to covariance windup when data are rank deficient, hence blowup problems are inherently avoided.

2. The forgetting factor refers to the MSL of the information content of each individual parameter in the system, hence it is a less sensitive, but more meaningful tuning knob than those proposed for exponential forgetting in Chapter 5.

It is recommended that directional forgetting be used in new adaptive control applications. In view of the two main advantages this algorithm has over exponential forgetting, it is recommended that current applications which use exponential forgetting be modified to use directional forgetting if problems of covariance windup or adaptivity are experienced.

Chapter 7

Application of Least Squares Type Methods to Bounded Disturbances

7.1 Introduction

Chapters 2 and 3 describe methods of modelling and identification which are based on the assumption that the disturbances are modeled as filtered white noise. It is often the case that the statistics of the noise can not be described by the linear filter methods presented in Chapter 2. A more general approach to describing disturbances (Peterson and Narendra, 1982) is to assume that they are bounded within known limits. This approach is often useful to describe situations where the disturbance properties are complex or non-linear. In these cases however, the arguments made in deriving methods based on the "minimisation of prediction errors" are not valid (Fogel and Huang, 1982). A classic example of bounded noise is the finite accuracy or quantisation error which occurs in digital sampling and finite word length arithmetic.

The bounded noise description is useful in the formulation of fault diagnosis problems where the disturbance corresponds to a fluctuation of the component values around their nominal values (Isermann, 1984). In these problems a fault is recognised when the disturbance estimate falls outside the pre-specified bounds.

An interesting geometrical interpretation of the bounded disturbance type algorithms results from the graphical analysis of the P matrix as described in Chapter 5. The bounded noise disturbance model allows the identification problem to be interpreted from a "set theoretic" viewpoint, since the confidence bounds of the parameter estimates will be absolute, and form an exact closed set in parameter space.

Chapter 7 presents a brief survey of recent approaches to the problem of identification with the bounded disturbance assumption, and includes a critical review of a novel algorithm due to Dasgupta and Huang (1987) known as "modified least squares" (MLS).

The performance of MLS is directly compared to the algorithms of Chapter 5 by applying this algorithm to the distillation column data. A direct comparison is made between the optimally bounding ellipses of the MLS algorithm and the analogous confidence bound ellipses of the ordinary least squares type algorithms.

7.2 Systems with Bounded Disturbances

There have been numerous approaches to the problem of identification motivated by the assumption of bounded disturbances (Peterson and Narendra, 1982; Fogel and Huang, 1982; Goodwin and Sin, 1985). A substantial result in the theoretical analysis of the recursive least squares algorithm in the presence of bounded disturbance or "finite accuracy" data is a new version (Niederlinski, 1984b) of the now classic least squares convergence theorem (Ljung, 1976) which proves the property of consistency of the least squares estimates. Niederlinski (1984a) observed that the idea of a "true" parameter vector being a unique quantity which has infinite accuracy and may be approached with probability one is absurd, since it contradicts an "engineering axiom":

It is not possible to increase the accuracy of an estimate indefinitely by simply increasing the number of measurements.

The conclusion drawn from the analysis is that no matter how large the number of measurements, the least squares estimate will have an asymptotic uncertainty bound equal to the "mean squared error" of the measurement, and that after that bound is reached no improvement in the estimate is realised by additional observations. Niederlinski (1984a) concludes that shutting off the estimation algorithm when the error is within a certain bound is an intuitive and natural result which is supported from a theoretical standpoint.

7.3 The Prediction Error Dead Zone

It has been demonstrated (Goodwin and Sin, 1984) that shutting off an estimation algorithm when the prediction error is within a certain bound enhances estimator robustness and improves performance for systems with a wide range of noise structures. Consider that a system with a bounded disturbance is described by the following model:

$$y(t) = \phi^T(t)\theta_0 + v(t) \tag{7.1}$$

with the disturbance bound given by γ such that:

$$v(t)^2 \leq \gamma^2 \tag{7.2}$$

The dead-zone prediction error method resulting from this model (Goodwin, and Sin, 1984; Peterson and Narendra, 1982) is given as:

$$\hat{\vartheta}(t) = \hat{\vartheta}(t-1) + \begin{cases} K(t)(y(t) - \hat{\vartheta}^T(t-1)\phi(t)) & \epsilon>\delta \\ 0 & \epsilon\leq\delta \end{cases} \tag{7.3}$$

The dead zone δ is chosen to be twice the assumed error bound:

$$\delta=2\gamma$$

This choice results in prediction errors which asymptotically approach a bounded region twice the width of the assumed error bound (Goodwin and Sin, 1984):

$$\lim_{t\to\infty} \sup \epsilon(t) \leq 2\gamma \tag{7.4}$$

Larger dead zones imply that adaptation will take place over a shorter period of time, and in general, larger output and parameter errors will necessarily result (Peterson and Narendra, 1982).

A clear disadvantage to the simple dead zone modification given by (7.3) is that if γ is overestimated then the prediction error will have limiting values no smaller than twice the assumed bound. The modified least squares algorithm (Dasgupta and Huang, 1987) introduces a time varying dead zone which has been proposed as a potential solution to this limitation. This algorithm is discussed further in Section 7.5.

7.4 Experimental Evaluation of the Dead Zone Modification

Some practical insight into the value of the dead zone modifications may be drawn from applying such an algorithm to a data set containing process noise. Such a study was undertaken using the distillation column data set described in Chapter 5, in which ordinary recursive least squares with a dead zone was applied to the data. The experiment was repeated ten times with values of γ varying from 0.0 to 0.3. The accuracy of the parameters was measured by V, the sum of the prediction errors squared:

$$V = \sum_{t=1}^{N}(y(t) - \hat{\theta}^T(t)\phi(t))^2$$

These values are plotted against the size of the dead zone in Figure 7.1a. During each run the fraction of the total number of data values which was actually used for estimation was recorded, and these values are plotted against dead zone in Figure 7.1b.

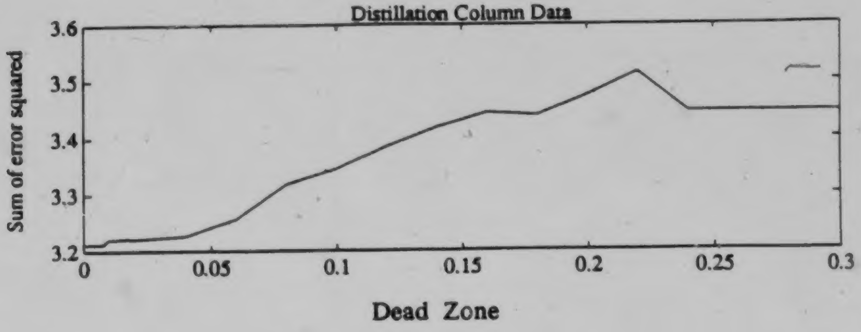

Figure 7.1 a
The sum of the prediction error squared versus the width of the dead zone for the application of RLS to the distillation column data.

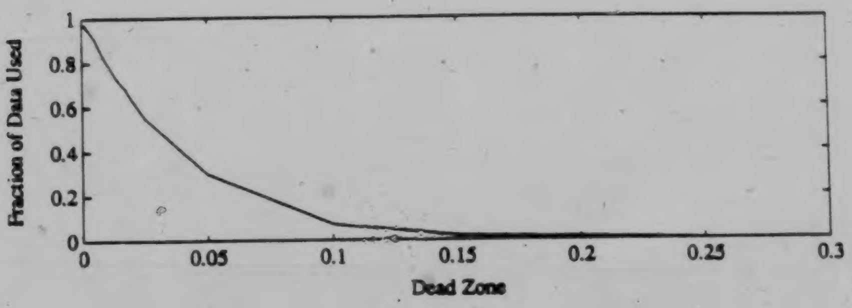

Figure 7.1 B
The fraction of the total data used versus the width of the dead zone for the appliction of RLS to the distillation column data.

For $\gamma > 0.25$ no updating occurred, hence that value of $V = 3.445$ corresponds to the accuracy of the initial parameter estimates only. As γ was decreased to the range 0.2-0.25 only a few data values are used for estimation, however an unexpected result occurs. The estimation algorithm used the data values with the largest prediction errors, and the value of V increased, rather than decreased. In this case the data points with the largest prediction errors are probably outliers due to unusual disturbances or non-linearities. These have an obvious adverse effect on the parameters, and it is interesting to note that larger values of the dead zone actually increases the influence of these points.

As γ continues to decrease more data is used for parameter estimation, and the accuracy improves. By examining the slope of the V curve it appears that the most valuable data occurs in the dead zone range 0.05-0.15, although there is a diminishing return on accuracy that is achieved.

little improvement in accuracy is achieved by lowering the dead zone below 0.05, even though less than half of the data in the set has been used at that point

From this example two valuable observations may be made relating to the practical application of an estimation dead zone:

1. Decreasing the dead zone will increase the fraction of data used in the estimation, but will have a diminishing return on the accuracy of the estimation.

2. The influence of outliers on the estimation error is increased as the dead zone is increased.

The first observation is consistent with that of Neiderlinski (1984a), in that the value of γ may increase over a fairly wide range without having a strong impact on accuracy. Increasing γ is generally beneficial because it reduces computational load and variation in the parameters. The second observation however, implies that choosing a value of γ which is too large can have a very detrimental effect on parameter estimation because of the increased influence of outliers. Clearly the value of the dead zone is an important parameter which should be chosen based on consideration of these two effects.

The subject of dealing with outliers is beyond the scope of this work, however its importance and impact on recursive estimation should not be overlooked, and the similarity of the techniques to those proposed for handling bounded disturbances make it worthy of mention. There have been different methods proposed for the elimination of outliers by fault detection techniques. These generally detect an outlier by some form of statistical confidence test applied to the current observation. The resulting algorithms reject a data value when the prediction error is too large. A survey of these is presented in Isermann (1984). Ljung and Soderstrom (1983) use a different approach for dealing with outliers which uses a modified criterion function to reduce the weighting placed on data values associated with large prediction errors.

The idea of a bounded disturbance model motivates an interesting and useful geometric interpretation of identification (Fogel and Huang, 1982) which is quite analogous to the confidence ellipsoid discussed in Chapter 5. The following section describes this geometric interpretation, employing a "membership set" concept as an analogy of the covariance bounds of the parameter estimate.

7.5 A Set Theoretic Approach to Parameter Estimation

Recursive identification algorithms applied to bounded disturbance models may be formulated as set membership problems in which the size and boundaries of the membership set are sequentially updated, rather than a covariance matrix as is used in the ordinary least squares type methods. This idea has been explored by numerous authors (Schweppe, 1971; Fogel and Huang, 1982; Niederlinski, 1984a; Norton 1987, Lozano-Leal and Ortega, 1987; Dasgupta and Huang, 1987).

In this approach the parameter membership set represents a bounded volume in parameter space which contains the intersection of the regions defined by all of the previous observations. The true parameters θ_0 should always lie within that membership set, hence there is an incentive to reduce the size of the set as efficiently as possible with the data available.

The set membership identification problem (Fogel and Huang, 1982) may be re-stated as follows:

Given the data $\left\{ y(t),\ \phi(t),\ \gamma;\ t=1,\ldots k \right\}$ *find a set* Θ *in* $\mathbb{R}^n$ *which is consistent with the system description given by equations* (7.1) *and* (7.2).

The region Θ is the membership set of the estimates of the unknown vector θ_0. The best or optimal set denoted by Θ_0 is given by the intersection of all the constraints S(t) imposed by the disturbance bounds of each data point:

$$\Theta_0(t) = \bigcap_{i=1}^{t} S(i) \tag{7.5}$$

Any other set $\Theta(t)$ which is consistent with (7.1) and (7.2) for each observation must also satisfy:

$$\Theta_0(t) \subset \Theta(t) \tag{7.6}$$

An expression for the polytope (region bounded by a set of planes in $\mathbb{R}^n$) S(t) is given by combining equations (7.1) and (7.2):

$$(y(t)-\theta_0^T\phi(t))^2 \leq \gamma^2 \tag{7.7}$$

The polytope S(t) is defined as:

$$S(t) = \left\{ \theta: (y(t)-\theta^T\phi(t))^2 \leq \gamma^2 \ ; \theta \in \mathbb{R}^n \right\} \tag{7.8}$$

The regions defined by S(t) are formally termed "degenerate ellipsoids" in $\mathbb{R}^n$ (Fogel and Huang, 1982). Geometrically, the S(t) region is the space bounded by two parallel hyperplanes separated by a distance given by $2\gamma/\|\phi(t)\|$. Each region S(t) must contain the true parameters θ_0 between the hyperplanes hence the intersection of many such regions will define a volume which also encloses θ_0. Note that the volume will be closed if and only if the set of vectors $\phi(t)$ span n dimensions.

The problem of determining the tightly bounded set given by (7.5) is intractable for large values of t because it involves the solution of 2·N n^{th} order inequalities. A reformulation of the problem was posed (Fogel and Huang, 1982) which finds a set $\Theta(t)$ which tightly contains Θ_0 (subject to (7.6)) and is computationally simple to evaluate. The simplest set satisfying (7.6) is mathematically derived (Fogel and Huang, 1982) from the weighted sum of the S(t) polytopes:

$$\Theta(t) = \left\{\theta: \sum_{k=1}^{t} \alpha_k(y(k)-\theta^T\phi(k))^2/\gamma^2 \leq \sum_{k=1}^{t} \alpha_k ; \ \alpha_k \geq 0\right\} \tag{7.9}$$

This equation is similar to the least squares cost function given by (3.17), however its solution is not given by a single value of $\hat{\theta}(t)$, as in ordinary least squares. A solution to (7.9), due to Dasgupta and Huang (1987) is an ellipsoidal volume in $\mathbb{R}^n$ given by:

$$\Theta(t) = \left\{ \theta: (\theta-\hat{\theta}(t))^T P^{-1}(t)(\theta-\hat{\theta}(t))/\gamma^2 \leq \sigma^2(t) \right\}$$

where:

$$P^{-1}(t) = \sum_{k=1}^{t} \alpha_k \phi(k)\phi^T(k) \text{ and } \sigma^2(t) = \sum_{k=1}^{t} \alpha_k y^2$$

The data weighting factor α_k is clearly analogous to that used in the development of ordinary least squares in Chapter 3. The centre of the ellipsoid corresponds to the weighted least squares estimate (given by equation (3.17)). The problem of recursively finding $\Theta(t)$ is reduced to choosing the value of the weighting sequence α_i. The weighting factor introduces an additional degree of freedom to the resulting algorithm which may be used to satisfy any other requirements which are imposed on the problem.

Figure 7.2 illustrates the intersection of an elliptical set $\Theta(t-1)$ with a typical data value $S(t)$. In this case the polytope $S(t)$ is simply a pair of lines perpendicular to the vector $\phi(t)$, separated by a distance $2\gamma / \|\phi(t)\|$, with θ_2 axis intercepts of $(y(t) \pm \gamma)/\phi_2$.

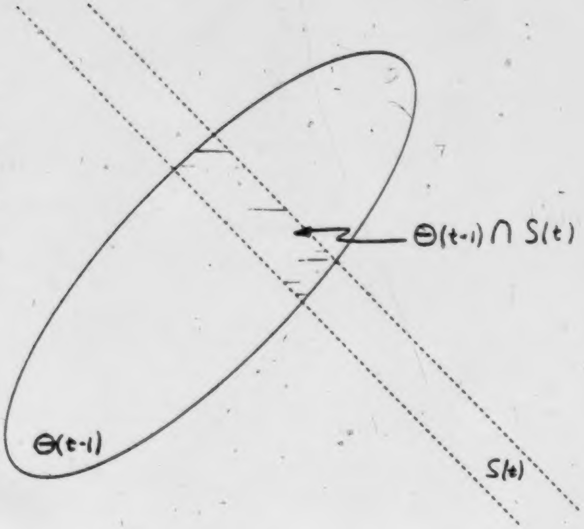

Figure 7.2 Intersection of Θ(t-1) with S(t)
The intersection between the region defined by Θ(t-1) (solid line) and S(t) (broken lines) is shaded.

7.6 The Modified Least Squares Method

Most recursive algorithms are designed to continually update the parameters without regard to the benefits provided, or to merely shut off the update when an arbitrary criterion such as the prediction error falls within certain bounds. The MLS algorithm of Dasgupta and Huang (1987), however has been shown to possess superior properties to standard dead-zone type modifications, due to its incorporation of "information dependent" updating and exponential forgetting. Asymptotic cessation of parameter updates and parameter convergence has been proven (Dasgupta and Huang, 1987) under the assumption that the input sequence is persistently exciting.

While the prediction error for standard dead-zone algorithms has limiting values given by twice the assumed noise bound, it was demonstrated (Dasgupta and Huang, 1987) that with even up to 20 percent overestimation of γ, the prediction error asymptotically approaches values smaller than the actual noise bound.

Consider a bounding ellipsoid $\Lambda(t-1)$ that is sufficiently large to cover all the likely values of θ_0, given by:

$$\Lambda(t-1) = \left\{\theta:(\theta-\hat{\theta}(t-1))^T P^{-1}(t-1)(\theta-\hat{\theta}(t-1)) \leq \mathcal{E}^2(t-1)\right\} \tag{7.10}$$

Where $\mathcal{E}^2(t-1)$ is a scalar parameter error function which provides an upper bound on the Lyapunov function $V(t-1)$, defined by:

$$V(t-1) = (\theta_0-\hat{\theta}(t-1))^T P^{-1}(t-1)(\theta_0-\hat{\theta}(t-1)) \tag{7.11}$$

$$\mathcal{E}^2(t-1) \geq V(t-1) > 0 \text{ iff } \theta_0 \in \Lambda(t-1)$$

When the data point $[y(t),\phi(t)]$ is acquired at each interval the algorithm finds an ellipsoid $\Lambda(t)$ which bounds the intersection of the ellipsoid $\Lambda(t-1)$ and the polytope $S(t)$, which is in some sense optimal. Such an ellipsoid is given by the linear combination of (7.10) and (7.8) as:

$$\begin{aligned}\Lambda(t) = \Big\{\theta:(1-\beta(t))(\theta-\hat{\theta}(t-1))^T P^{-1}(\theta-\hat{\theta}(t-1)) + \beta(t)(y(t)-\theta^T\phi(t))^2 \\ \leq (1-\beta(t))\mathcal{E}^2(t-1) + \beta(t)\gamma^2\Big\}\end{aligned} \tag{7.12}$$

The data weighting parameter, $\beta(t)$, is chosen to satisfy $\beta\in[0,\beta_{max}]$; $\beta_{max}<1$. An illustration of its effect on the size and shape of the ellipse $\Lambda(t)$ is depicted in Figure 7.3. Note that the region defined by the intersection of the ellipse $\Lambda(t-1)$ and $S(t)$ is completely enclosed by $\Lambda(t)$, for all $\beta(t)$.

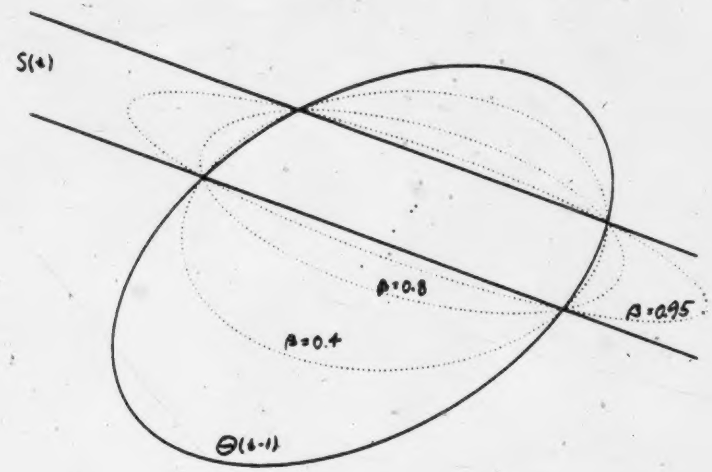

Figure 7.3 Effect of β Weighting on the Bounding Ellipse
The resulting region $\Theta(t)$ may take a range of sizes and shapes depending on the data weighting $\beta(t)$. Ellipses for $\beta=0.4$, $\beta=0.8$, $\beta=0.95$ are drawn in dotted lines.

The algorithm due to Dasgupta and Huang (1987) for the update step $\Lambda(t-1) \rightarrow \Lambda(t)$ which guarantees the existance of $\delta^2(t)$ and $P(t)$ such that:

$$\Lambda(t) = \left\{\theta:(\theta-\hat{\theta}(t))^T P^{-1}(t)(\theta-\hat{\theta}(t)) \leq \delta^2(t)\right\} \tag{7.13}$$

is given as:

$$P^{-1}(t) = (1-\beta(t))P^{-1}(t-1) + \beta(t)\phi(t)\phi^T(t) \qquad (7.14)$$

$$\hat{\vartheta}(t) = \hat{\vartheta}(t-1) + \beta(t)P(t)\phi(t)\epsilon(t) \qquad (7.15)$$

$$\mathcal{E}^2(t) = (1-\beta(t))\mathcal{E}^2(t-1) + \beta(t)\gamma^2 - \frac{\beta(t)(1-\beta(t))\epsilon^2(t)}{1-\beta(t) + \beta(t)G(t)} \qquad (7.16)$$

with:

$$\epsilon(t) = y(t) - \hat{\vartheta}^T(t-1)\phi(t) \qquad (7.17)$$

The value of $\beta(t)$ is calculated in order to minimise $\mathcal{E}^2(t)$ with respect to $\mathcal{E}^2(t-1)$, hence "optimally" bounding the Lyapunov function V(t) given by (7.11). This strategy is based on the need to minimise the extent of the "feasible" parameter estimate (measured by $\mathcal{E}^2(t)$), in contrast to other variable forgetting factors (such as those presented in Chapter 5) which are based on the desire to maintain some measure of the magnitude or information content of P at a constant level. The details of the calculation of $\beta(t)$ are presented elsewhere (Dasgupta and Huang, 1987). The update equations (7.14)-(7.16) are similar to the ordinary least squares algorithm with an exponential forgetting factor. The weighting series α_i; $i \le t$ is related to $\beta(t)$ as:

$$\alpha_t = \beta(t);\ \alpha_{t-1} = \beta(t-1)(1-\beta(t));\ldots\alpha_i = \beta(i)\prod_{j=i+1}^{t}(1-\beta(j)) \qquad (7.18)$$

This sequence is directly related to the forgetting factor defined by equation (5.4), with the value of $\lambda(t)$ given by $(1-\beta(t))$. The key difference between this and the ordinary forgetting factor algorithms is that the weighting at time t, α_t is chosen as $\beta(t)$, rather than unity, hence $\beta(t)$ appears as a gain factor in the update equations. Making this modification to the weighting sequence gives the algorithm a built-in dead zone because the updating is shut-off when the weighting parameter $\beta(t)$ is chosen as zero.

The convergence proof presented for the MLS algorithm (Dasgupta and Huang, 1987) assumes that $\phi(t)$ and G(t) are bounded, and that the data matrix P is of full rank. The proof demonstrates that the parameter error converges to a region in which:

$$\hat{\theta}(t) - \theta_0{}^2 \leq \gamma^2/d_1 \tag{7.19}$$

where:

$$0 < d_1 I \leq P^{-1}(t) \leq d_2 I \tag{7.20}$$

The parameter error is bounded by a value which is directly proportional to the noise bound γ^2 and inversely proportional to the minimum eigenvalue of P^{-1}.

An important aspect of the algorithm is the interpretation of the parameter error bound $\mathcal{E}(t)$. If the true parameters lie within the bounding ellipsoid $\Lambda(t)$ at time t then $\mathcal{E}(t)$ will remain a non-increasing positive value which provides an upper bound on a Lyapunov function given by:

$$V(t) = (\theta_0 - \hat{\theta}(t))^T P^{-1}(t)(\theta_0 - \hat{\theta}(t)) \tag{7.21}$$

$$\delta^2(t) \geq V(t) \quad \text{iff } \theta_0 \in \Lambda(t) \tag{7.22}$$

The asymptotic cessation of parameter updating has been theoretically proven (Dagupta and Huang, 1987) based on this result.

7.7 Experimental Evaluation

A PC-Matlab implementation of the MLS algorithm was developed and is included in Appendix B. This implementation was applied to the distillation column data described in Chapter 5 as evaluation of its practicality under realistic conditions.

The algorithm was applied to the estimation of the two parameters of an ARIMA model given by:

$$\Delta y(t) = -a_1 \Delta y(t-1) + b_0 \Delta u(t-1) \tag{7.23}$$

In order to investigate the sensitivity to the estimate of the noise bound γ, the test was repeated three times with different values of γ. From Figure 7.1b it is estimated that the actual bound will be in the range of 0.18-0.25 hence the values $\gamma^2 = 0.07$, 0.04 and 0.03 were used.

The parameter trajectories for each case are plotted in Figure 7.4 a,b and c. The weighting factors $\beta(t)$ for each run are plotted in Figures 7.5 a,b and c and the Lyapunov bound $\delta^2(t)$ is plotted in Figure 7.6 a,b and c.

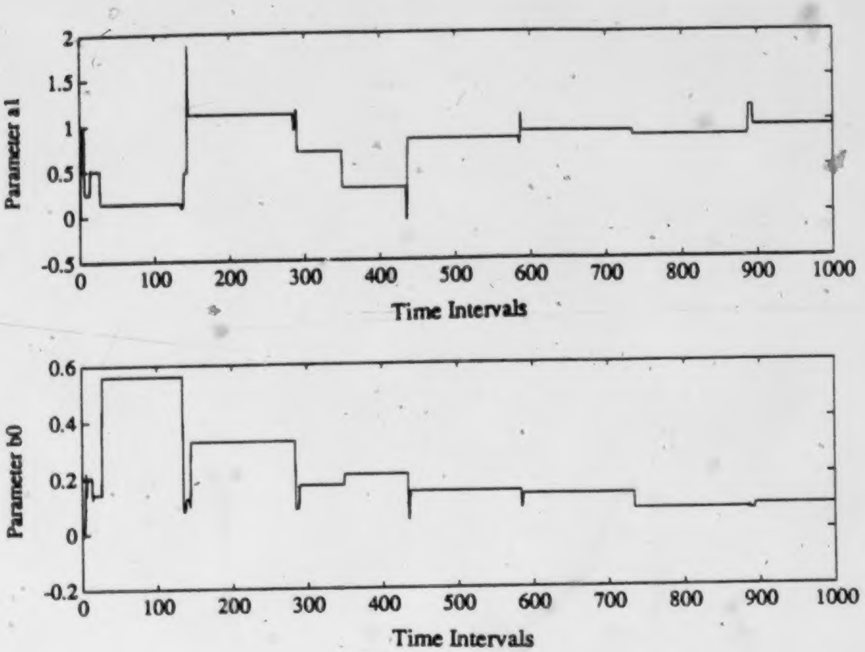

Figure 7.4 a

Parameter estimates - Modified Least Squares Algorithm applied to Distillation Column data with γ^2=0.07

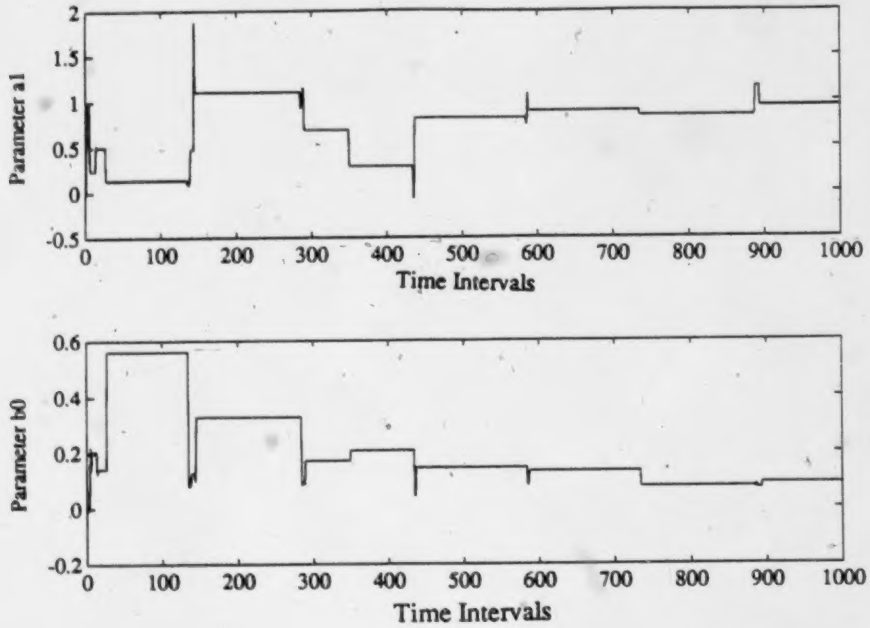

Figure 7.4 b

Parameter estimates - Modified Least Squares Algorithm applied to Distillation Column data with γ^2=0.04

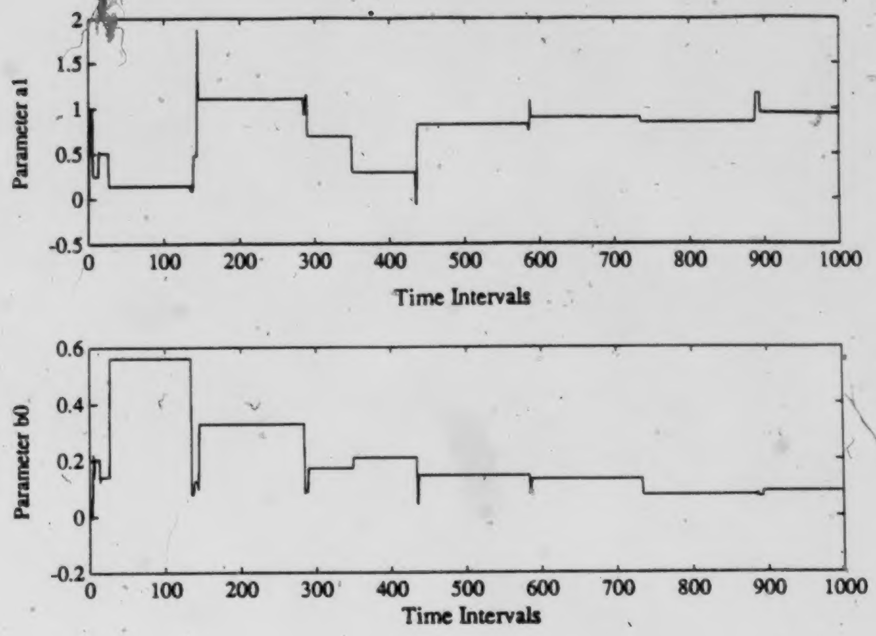

Figure 7.4 c

Parameter estimates - Modified Least Squares Algorithm applied to Distillation Column data with γ^2=0.03

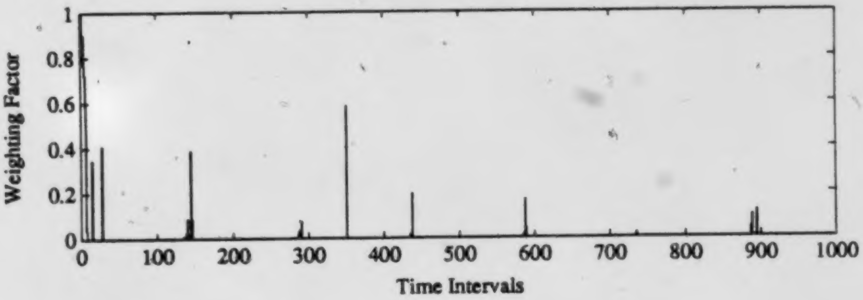

Figure 7.5a

Weighting factors $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with γ^2=0.07

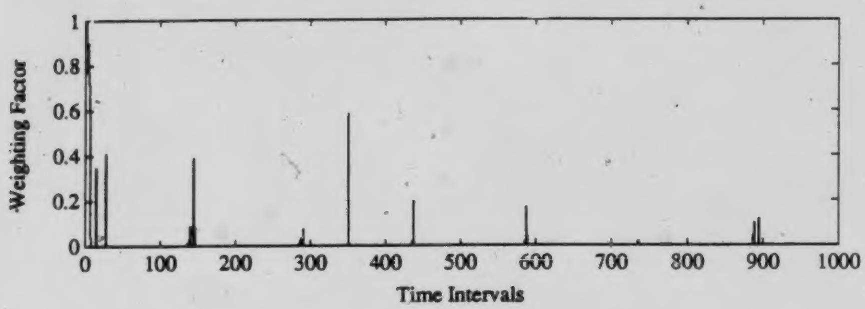

Figure 7.5b
Weighting factors $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with $\gamma^2=0.04$

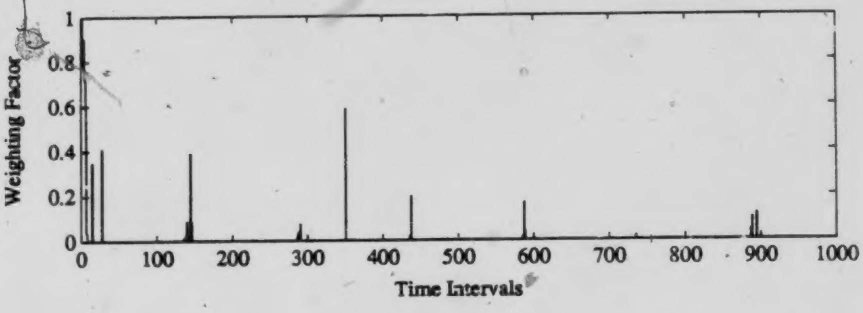

Figure 7.5c
Weighting factors $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with $\gamma^2=0.03$

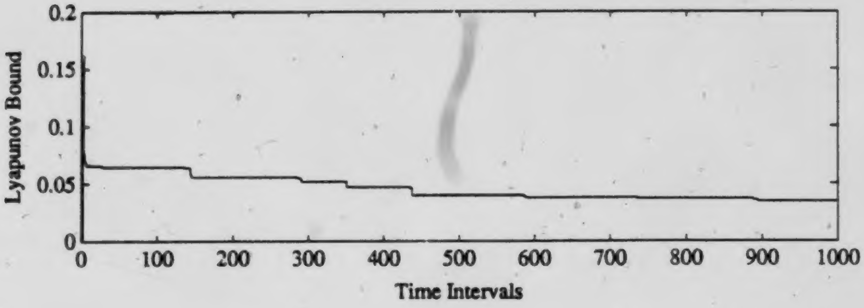

Figure 7.6a
Lyapunov Bound $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with $\gamma^2=0.07$

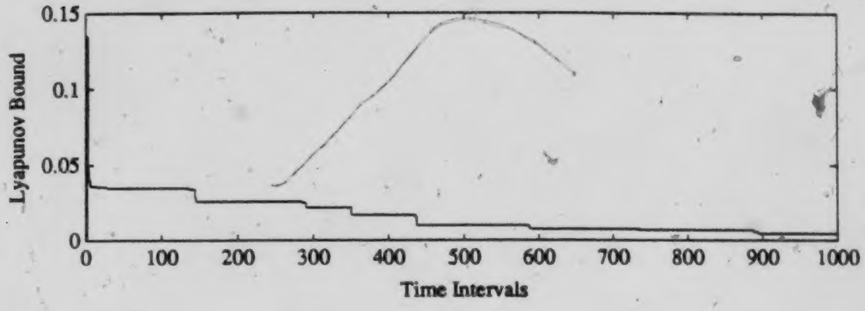

Figure 7.6b
Lyapunov Bound $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with $\gamma^2=0.04$

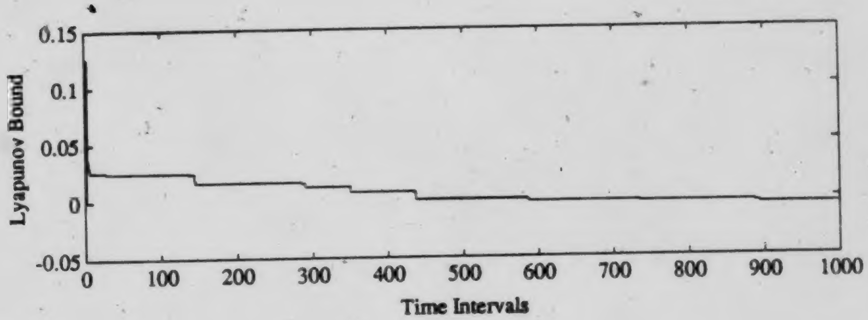

Figure 7.6c
Lyapunov Bound $\beta(t)$ - Modified Least Squares Algorithm applied to Distillation Column data with $\gamma^2=0.03$

7.8 Observations on the MLS Algorithm

Figures 7.4a,b,c indicate remarkably different behavior of this algorithm over those previously examined in Chapters 5 and 6. After an initial convergence period the updates in the parameter are infrequent, occuring only with setpoint changes. The final values of the parameter estimates are very close to those achieved by the more conventional estimation methods in Chapters 5 and 6, although in the initial 500 intervals the variation in the estimates was very large.

The large parameter variation was clearly due to the form of equations

(7.14),(7.15) and (7.16) which use $\beta(t)$ as both a data weight and an exponential forgetting factor. As a result a great deal of weight is placed on those points with large prediction errors, and elimination of outliers is much more important than in the usual least squares type algorithms.

A critical weakness in the algorithm is revealed when the value of the assumed bound γ^2 is chosen smaller than the actual bound. Comparing Figures 7.6a,b,c it is apparent that γ^2 directly influences the asymptotic value of ε^2. In Figure 7.6c, in which $\gamma^2 = 0.03$, the value of ε^2 actually drops below zero. This is an irrational result which precedes total numerical failure of the algorithm. The algorithm is therefore intolerant to underestimation of the noise bound, and outliers. The non-increasing nature of ε^2 is a potential numerical problem which limits the applicability of this algorithm.

Comparing the algorithm for γ^2=0.07 to γ^2=0.04 it is clear that moderate overestimation of the noise bound apparently does not adversely affect the resulting parameter estimates, and the value of ε^2 remains positive.

7.9 Summary

The bounded disturbance description is a very practical and general method for modelling system noise and disturbances. The idea of shutting off the estimation when the error is smaller than some lower bound is an intuitive and natural modification which results from this description.

An experimental evaluation of this modification was made by applying the RLS algorithm to the distillation column data of Chapter 5, and varying the width of a prediction error dead-zone. From this it is concluded that decreasing the width of the dead zone will increase the amount of data used for identification, but will provide a diminishing return on the accuracy of the estimation. Furthermore it was observed that the influence of outliers on the estimation error is increased as the dead zone is increased. It is concluded that the width of the dead zone is an important parameter which should be chosen based on consideration of these two effects.

The bounded description model allows the estimation algorithm to be constructed as a membership set problem. This idea has motivated numerous approaches to recursive identification with the bounded noise description (Schweppe, 1971; Fogel and Huang, 1982; Niederlinski 1984a; Norton, 1987; Lozano-Leal and Ortega, 1987; Dasgupta and Huang, 1987). The analogy of the membership set to the covariance bound ellipse is illustrated and discussed.

The MLS algorithm of Dasgupta and Huang (1987) was evaluated by application to the distillation column data of Chapter 5. The algorithm has several advantages over the ordinary application of a dead zone modification because it uses a time varying dead zone which makes efficient use of data and allows some over-estimation of the dead zone without significant penalty. It is observed that the algorithm is very sensitive to under-estimation of the noise bounds, and that this generally leads to numerical problems.

A general observation that may be made of the MLS algorithm is that uses the available information very efficiently, since very few of the data values are actually used for estimation. Its large parameter variation and numerical intolerance of outliers, however place a great restriction on the general applicability of the algorithm in its present form. It is recommended that further development of the algorithm be undertaken to improve its robustness, and reduce its sensitivity to underestimation of γ.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

A broad study of various aspects of recursive estimation has been conducted, covering the topics of choice of model structure, choice of estimation algorithm, use of the differencing operator for treatment of non-zero steady states, exponential forgetting, directional forgetting, and treatment of the bounded disturbance description. The main conclusions of the thesis are summarised as follows:

A common problem in modeling real systems is that the input and output data have nominal, or steady state values which are non-zero. Two approaches to the problem have been identified:

1. Pre-treat the data by subtracting out estimates of the steady states, and re-defining the input and output of the model in terms of "deviations" from real or estimated physical equilibria.

2. Write the model in terms of the actual or "positional" data values by lumping the steady state values in a displacement term μ. The displacement term may appear explicitly in the model or be considered part of a non-zero mean disturbance.

Model structures in wide use generally fall into one of two main classes based on different assumptions of how the disturbance affects the output. The equation error description which assumes the disturbance to be an extra input to the system, hence the process and disturbance models will have common poles (given by the $A(q^{-1})$ polynomial), and the output error description which assumes that the disturbance acts upon the measurement of the output by an independent mechanism. The choice of a model structure from one of these two broad classes should ideally be based on how well the physical situation conforms to the structure of the model.

The equation error forms are generally preferred for recursive applications because the identification methods available for these are simpler to implement and convergence is faster.

The many variations available for choice of model structure described in Chapter 2 generate a need for "customised" approaches to the recursive parameter identification problem. Recursive prediction error methods have been widely studied in the literature, and discussed in Chapter 3. A comprehensive treatment of the subject has been developed in the form of numerous algorithms, each of which may be interpreted as a variation or extension of the basic recursive least squares method.

In general, the more complicated identification algorithms (eg. GRML, PLR, IV) should be avoided in favor of RLS whenever possible because for process data problems the properties of the disturbance are often not uniform, and the noise models are difficult to estimate by recursive methods. In choosing the model to be identified it is always advisable to incorporate as much *a priori* knowledge of the system as possible in order to reduce the number of parameters to be identified. The integrating noise model is a special application of *a priori* knowledge to the disturbance model.

It is demonstrated that a basic admissibility condition on the C polynomial is that all of its roots must remain in the right half of the z plane. This is based on the observation that the ARIMAX model results from an unfiltered Brownian motion process plus a filtered white noise signal.

In Section 4.5 it was demonstrated that the C polynomial of the ARIMAX structure is mathematically equivalent to a Kalman filter for tracking the value of the displacement term μ. The mathematical equivalence of the equations is proven and a numerical example using the RML and ELS methods was presented to demonstrate the practicality of this result. From the example considered it is concluded that the ELS method, although less rigorously supported in theory than the RML method gives superior performance with respect to rate of convergence. No apparent explanation is available for this result, hence it is suggested as a topic for future investigation.

A new exponential forgetting factor which is based on maintaining the determinant of the P matrix equal to a constant is derived and evaluated. It is found by simulation and by using experimental data that this approach is robust since it causes the forgetting factor to go to unity when the data is not persistently excited.

A comparison of this method to a constant trace (Sripada and Fisher, 1987) and a constant information forgetting factor (Fortescue *et al.*, 1981) was made using simulated and real process data. The constant trace and constant determinant were found to behave quite similarly. It was concluded that the constant information forgetting factor however was less robust because it was found to cause covariance windup due to process noise.

The confidence bound ellipse is presented as a valuable tool for graphically examining a number of fundamental properties of the P matrix. The eigenvalues, eigenvectors, trace, determinant and condition number of a 2×2 matrix may be interpreted by examination of the ellipse. The ellipse is found to be useful in interpreting the various approaches to data forgetting.

The directional forgetting algorithm (Kulhavy and Karny, 1984) was found to have at least two distinct advantages over exponential forgetting:

1. The method is not subject to covariance windup when data is rank deficient, hence blowup problems are inherently avoided.

2. The forgetting factor refers to the MSL of the information content of each individual parameter in the system, hence it is a less sensitive, but more meaningful tuning knob than those proposed for exponential forgetting in Chapter 5.

An important observation made of the directional forgetting algorithm is that the value of the forgetting factor is not related the number of parameters in the identification problem. This contrasts with the ordinary exponential forgetting factor which, in general must increase to the power of n.

The bounded disturbance description is a very general and practical approach to the problem of dealing with disturbances in recursive identification. The idea of shutting off the estimation when the error is smaller than some lower dead-zone bound is an intuitive and natural modification which results from this description. An experimental evaluation of this modification is made by applying an RLS algorithm to the distillation column data of Chapter 5, and varying the width of the prediction error dead zone. From this it is concluded that decreasing the width of the dead zone will increase the amount of data used for identification, but will provide a diminishing return on the accuracy of the estimation. Furthermore it was observed that the influence of outliers on the estimation error is increased as the dead zone is increased. It is concluded that the width of the dead zone is an important parameter which should be chosen based on consideration of these two effects.

It was concluded based on an application of the MLS algorithm of Dasgupta and Huang (1987) to the distillation column data that the algorithm uses available information very efficiently, and it allows moderate overestimation of the assumed disturbance bound. Its large parameter variation and numerical intolerance of outliers, however limits its applicability.

It may be concluded that the field of recursive estimation is not yet fully explored but still evolving, judging from the number of new ideas and innovations that have appeared in the last five years. In the current state, however there are remarkably few problems which remain to be solved. The acceptance and adoption of this technology in industry in the form of adaptive control is still not at all common. This is perhaps inhibited by past failures and a generally poor understanding of the latest techniques available.

8.2 Recommendations for Future Work

Several areas of future study in the area of recursive identification and adaptive control may be considered.

- A comparative study of the numerical aspects of the ELS and the RML algorithms is suggested to provide an explanation for the results of the simulated example presented in Chapter 4.

- The convergence properties of recursive least squares with a constant determinant forgetting factor should be investigated.

- The convergence properties of recursive least squares with directional forgetting should be investigated.

- Experimental and industrial trials of the constant determinant, as well as the directional forgetting algorithms should be conducted in order to further investigate any practical problems in their use.

- The main problem with the "one in the data vector method" is that explicit estimation of μ adversely affects estimation of other parameters when there are load disturbances. This problem is at least partially solved by the use of directional forgetting, and it is therefore recommended that this idea be investigated further.

- The MLS algorithm of Dasgupta and Huang (1985) should be re-formulated to increase its robustness to underestimation of the noise bound.

- The confidence ellipse analysis provides a useful basis for evaluating the statistical significance of a measurement. Its uses in the field of fault detection and elimination of outliers should be further investigated.

BLANK PAGE INSERTED

References

Åström K.J., *Introduction to Stochastic Control Theory* Academic Press, New York (1970)

Åström K.J., "Maximum Likelihood and Prediction Error Methods" *Automatica* Vol.16, pp.551-574. (1980)

Åström K.J. and Bohlin, T.,"Numerical Identification of Linear Dynamic Systems From Normal Operating Records", *IFAC Symposium on Self-Adaptive Systems*, Teddington, England (1965)

Åström, K.J. and Söderström T., "Uniqueness of the Maximum Likelihood Estimates of the Parameters of an ARMA Model" *IEEE Transactions on Automatic Control*, Vol.AC-19, pp. 769-773, (1974)

Åström K.J. and Wittenmark B., "On Self Tuning Regulators",*Automatica*, Vol.9, pp.185-199. (1973)

Åstrom K.J., and Wittenmark B., *Computer Controlled Systems*. Prentice-Hall, Englewood Cliffs, N.J. (1984)

Bohlin T., "On the Problem of Ambiguities in Maximum Likelihood Identification" *IEEE Trans. Automatic Control*, Vol.AC-15, pp. 104-106 (1971)

Box G.E.P. and Jenkins G.M., *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco, (1970)

Clarke D.W., "Generalized Least Squares Estimation of Parameters of a Dynamic Model", *First IFAC Symposium on Identification in Automatic Control Systems*, Prague, Czechoslovakia (1967)

Clarke D.W., Hodgson A.J.F. and Tuffs P.S., "Offset problem and K-Incremental Predictors in Self-Tuning Control", *IEE Proceedings*, Vol.130, Part D, No.5, pp.217-225 (1983)

Cluett W.R., *Stable Adaptive Control in the Presence of Unmodeled Dynamics* Ph.D. Thesis, University of Alberta, (1986)

Cochrane D. and Orcutt G.H., "Application of Least Squares Regression to Relationships Containing Autocorrelated Error Terms", *J. Amer. Statist. Assoc.*, Vol. 44, pp. 32-61 (1949)

Dasgupta S. and Huang Y.F., "Asymptotically Convergent Modified Recursive Least-Squares with Data Dependent Updating and Forgetting Factor for Systems with Bounded Noise", *IEEE Transactions on Information Theory*, Vol.IT-33, No.3, pp.383-392 May (1987)

Finigan B. and Rowe I.H., "Strongly Consistent Parameter Estimation by the Introduction of Strong Instrumental Variables", *IEEE Trans. on Automatic Control*, Vol. AC-19, pp.825-831 (1974)

Fogel E. and Huang Y.F., "On the Value of Information in System Indentification-Bounded Noise Case", *Automatica* Vol.18, No.2, pp.229-238, (1982)

Fortescue T.R., Kershenbaum L.S. and Ydstie B.E., "Implementation of of Self Tuning Regulators with Variable Forgetting Factors", *Automatica*, Vol. 19,pp.831-835 (1981)

Francis B.A. and Wonham W.M., "The Internal Model Principle of Control Theory", *Automatica*, Vol.12, pp.457-465 (1976)

Goodwin G.C., Evans R.J., Lozano Leal R. and Feik R., "Sinusoidal Disturbance Rejection with Application to Helicopter Flight Data", *IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol. ASSP-34, pp. 479-484 (1986)

Goodwin G.C. and Sin K.S., *Adaptive Filtering Prediction and Control*, Prentice Hall, Englewood Cliffs N.J., (1984) Hastings-James R. and Sage M.W., "Recursive Generalized Least Squares Procedure for On-Line Identification of Process Parameters", *IEE Proc.*, Vol.116, pp.2057-2062. (1969)

Isermann, R., "Process Fault Detection Based on Modelling and Estimation Methods - A Survey", *Automatica*, Vol.29, No.4 pp.387-404, (1984)

Kulhavy R. and Karny M., "Tracking of Slowly Varying Parameters By Directional Forgetting" *Proc. 9th Triennial World Congress of IFAC*, Budapest Vol.2 pp.687-692. (1984)

Kulhavy R., "Restricted Exponential Forgetting in Real-Time Identification", *Automatica*, Vol. 23, No.5, pp.589-600, (1987)

Landau I.D., "Unbiased Recursive Estimation Using Model Reference Techniques", *IEEE Transactions on Automatic Control*, Vol.AC-21, pp.194-202. (1976)

Landau I.D. and Lozano R., "Unification of Discrete Time Explicit Model Reference Adaptive Controller Designs", *Automatica*, Vol.17, pp. 593-611. (1979)

Langman J., *Effect of Q Weighting on Self-Tuning Control of a Binary Distillation Column*, M.Sc. Thesis, University of Alberta, (1987)

Ljung L., *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, N.J. (1987)

Ljung L., "Comments on 'Non-convergence of the Approximate Maximum Likelihood Identification Algorithm'", *Automatica*, Vol.16, pp.219-220. (1980)

Ljung L., "Convergence Analysis of Parametric Identification Methods" *IEEE Transactions on Automatic Control*, Vol.AC-23,pp.770-783. (1978)

Ljung L., "Consistency of the Least Squares Identification Method" *IEEE Transactions on Automatic Control*, Vol.AC-21, pp.779-781 (1976)

Ljung L. and Söderström T., *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Massachusetts (1983)

Ljung L., Söderström T. and Gustavsson I., "Counterexamples to General Convergence of a Commonly Used Recursive Identification Method", *IEEE Transactions on Automatic Control*, AC-20 p. 643. (1975)

Ljung L., "Consistency of the Least Squares Identification Method", *IEEE Transactions on Automatic Control*, AC-21 pp.779-781. (1976)

Lozano-Leal R. and Ortega R., "Reformulation of the Parameter Identification Problem for Systems with Bounded Disturbances" *Automatica*, Vol.23, No.2, pp.247-251 (1987)

Niederlinski A., "A New Look at Least-Squares Dynamic System Identification", *Int. J. Control*, 40, No.3, pp.467-478. (1984a)

Niederlinski A., "Convergence of Least-Squares Dynamic System Identification with Finite Accuracy Data", *Int.J.Control*, 40, No.3, pp.479-485 (1984b)

Norton J.P., "Identification and Application of Bounded Parameter Models" *Automatica*, Vol.23,No. 4, pp.497-507. (1987)

Pacey W.C., *Control of a Binary Distillation Column: An Experimental Evaluation of Feedforward and Combined Feedforward-Feedback Control Schemes*, M.Sc. Thesis, University of Alberta, (1973)

Panuška V., "A Stochastic Approximation Method for Identification of Linear Systems Using Adaptive Filtering" *Proc. Joint Automatic Control Conference*, Ann Arbor Michigan. (1968)

Panuska V., "Non-Convergence of the Approximate Maximum Likelihood Identification Algorithm" *Automatica*, Vol.16, pp.215-217. (1980a)

Panuska V., "Authors Reply to Comments on Non-Convergence of the Approximate Maximum Likelihood Identification Algorithm" *Automatica*, Vol-16, p.221. (1980b)

Peterka V., "Bayesian Approach to System Identification", *Trends and Progress in System Identification*, Chapter 8 pp.239-304. Pergamon Press, Oxford (1981)

Peterson B.B. and Narendra K.S., "Bounded Error Adaptive Control", *IEEE Transactions on Automatic Control*, Vol.AC-27, pp.1161-1168 (1982)

Plackett R.L., "Some Theorems in Least Squares" *Biometrica*, Vol.37, p.149. (1950)

Robbins H. and Munro S., "A Stochastic Approximation Method" *Annals of Mathematical Statistics*, Vol.22,pp.400-407. (1951)

Schweppe F.C., "Recursive State Estimation: Unknown but Bounded Error and System Input", *IEEE Trans. Aut. Control*, AC-16, pp.117, (1971)

Shah S.L., *Parameter Estimation Techniques For Self-Tuning Control*, Research Report 861201 Department of Chemical Engineering, University of Alberta, Edmonton, Canada (1986)

Söderström T., "Convergence Properties of the Generalized Least Squares Identification Method" *Automatica* Vol.10, pp.617-626. (1974)

Söderström T., "On the Uniqueness of Maximum Likelihood Identification" *Automatica*, Vol.11 pp.193-197. (1975)

Solo V., "The Convergence of AML" *IEEE Transactions on Automatic Control*, Vol.AC-24,pp.958-963. (1979)

Song H.K., Shah S.L. and Fisher D.G., "Experimental Evaluation of a Robust Self-Tuning PID Controller", *Can. J. Chem. Eng.*, Vol. 62, pp. 755-763 (1984)

Sripada N.R. and Fisher D.G.,"Improved Least Squares Identifcation for Adaptive Controllers", *Proc. ACC*, pp.2027-2037 Minneapolis, (1987)

Svrcek W.Y., *Dynamic Response of a Binary Distillation Column*, Ph.D. Thesis, University of Alberta, (1967)

Tham M.T., Morris A.J., and Montague G.A., "Self-Tuning Process Control: a Review of Some Algorithms and their Application". *Proc. ACC*, Minneapolis (1987)

Tjokro S., *Derivation and Evaluation of an Adaptive PID Controller*. MSc. Thesis, University of Alberta (1984).

Tuffs P.S., "Self Tuning Control: Algorithms and Applications", OUEL report 1567/85 (1985)

Tuffs P.S. and Clarke D.W., "Self-Tuning Control of Offsets: A Unified Approach". *Proc. IEE*, Vol.132, No.3, pp. 100-110 (1985)

Vermeer P.J., Roffel B. and Chin P.A., "An Industrial Application of an Adaptive Algorithm to Overhead Composition Control", *Proc. of ACC*, (1987).

Vien A., "*Process Identification and Time Series Analysis*", Ph.D. Thesis, University of Alberta, (in preparation) (1989)

Wahlberg B. and Ljung L., "Design Variables for Bias Distribution in Trasfer Function Estimation", *IEEE Trans. Aut. Cont.* Vol. AC-31, No.2, pp. 134-144, (1986)

Wittenmark B. and Åström K.J., "Practical Issues in the Implementation of Self-Tuning Control". *Automatica*, Vol.20, No.5, pp. 595-605, (1984)

Wold H., *A study in the Analysis of Stationary Time Series*, Almqvist and Wiksell, Uppsala, Sweden 1938

Wong K.Y. and Polak E., "Identification of Linear Discrete Time Systems Using the Instrumental Variable Method", *IEEE Trans. Automatic Control*, Vol.AC-12, pp.707-718 (1967)

Ydstie B.E. and Co T., "Recursive Estimation with Adaptive Divergence Control", *IEE Proc.* pp. 124-130, Vol.132, Pt.D, No.3, (1985)

Young P.C., "An Instrumental Variable Method for Real-Time Identification of a Noisy Process", *Automatica*, Vol.6, pp.271-287 (1970)

Young P.C., "Some Observations on Instrumental Variable Methods of Time Series Analysis", *Int. Journal of Control*, Vol.23, pp.593-612 (1976)

Young P.C. and Jakeman A.J., "Refined Instrumental Variable Methods of Recursive Time-Series Analysis. Part I: Single Input Single Output Systems", *Intern. J. Control*, Vol.29, pp 1-30 (1979)

Young P.C., *Recursive Estimation and Time Series Analysis*. Springer-Verlag, New York (1984)

Appendix A

Proof of Theorem 1.1 Determinant of the Rank One Update

The determinant of the sum of an n×n positive definite matrix A^{-1}, multiplied by a scalar α and the outer product matrix xx^T (x is a vector with column dimension n) is given by the identity:

$$|\alpha A^{-1} + xx^T| = \frac{\alpha^{n-1}(\alpha + x^T A x)}{|A|} \tag{A.1}$$

Proof:

The proof is adapted from the derivation of the UDU^T factorisation of P(t) (Bierman, 1977).

Let:

$$A' = \left[\alpha A^{-1} + xx^T\right]^{-1} \tag{A.2}$$

Consider the factorisation:

$$A = UDU^T \tag{A.3}$$

The matrix U is upper triangular with all diagonal elements equal to unity and the matrix D is diagonal, therefore:

$$|A| = |U|\cdot|D|\cdot|U^T| = |D| \tag{A.4}$$

since $|U|=|U^T|=1$.

Defining the variables:

$$F = U^T x = \left[f_1 \; f_2 \; \ldots f_n \right]^T \tag{A.5a}$$

$$G = DF = \left[g_1 \; g_2 \; \ldots g_n \right]^T \tag{A.5b}$$

$$\beta_i = \alpha + \sum_{i=1}^{n} f_i g_i \tag{A.5c}$$

$$\beta_n = \alpha + x^T A x \tag{A.5d}$$

$$\beta_0 = \alpha \tag{A.5e}$$

Then from the matrix inversion lemma:

$$U'D'U'^T = \frac{1}{\alpha}\left[UDU^T - \frac{Ugg^TU^T}{\beta_n}\right] = \frac{1}{\alpha}U\left[D - \frac{gg^T}{\beta_n}\right]U^T \tag{A.6}$$

If the part in brackets may be further factorised as:

$$D - \frac{gg^T}{\beta_n} = \overline{UDU}^T \tag{A.7}$$

with:

$$U' = U\overline{U} \tag{A.8}$$

and

$$D' = \overline{D}/\alpha \tag{A.9}$$

then in order to find $|A'|$ from $|A|$ it is only necessary to find $\overline{D}$ in terms of D since:

$$|A| = |D'| = |\frac{1}{\alpha}\overline{D}| = \frac{1}{\alpha^n}|\overline{D}| \tag{A.10}$$

Introducing the notation:

$$\overline{U} = \left[\overline{U}_1 \ldots \overline{U}_n\right] = \begin{bmatrix} 1 & \overline{U}_{1,2} & \cdots & \overline{U}_{1,n} \\ & 1 & & \vdots \\ 0 & & \ddots & \overline{U}_{n-1,n} \\ & & & 1 \end{bmatrix} \tag{A.11}$$

$$\overline{D} = \begin{bmatrix} \overline{D}_1 & & 0 \\ & \ddots & \\ 0 & & \overline{D}_n \end{bmatrix} \tag{A.12}$$

$$D = \begin{bmatrix} D_1 & & 0 \\ & \ddots & \\ 0 & & D_n \end{bmatrix} \tag{A.13}$$

and let e_i be the unit vector of dimension *i*, then the factorization given by equation (A.7) implies:

$$\sum_{i=1}^{n} \overline{U}_i \overline{U}_i^T \overline{D}_i = \sum_{i=1}^{n} D_i e_i e_i^T - \frac{1}{\beta_n} GG^T \tag{A.14}$$

Introducing also:

$$V_n = G \tag{A.15}$$

and let the ith component of the column vector V_n is denoted by $V_{n,i}$, and:

$$V_{n-i} = \begin{bmatrix} V_{n,1} \\ \vdots \\ V_{n,n-i} \\ 0 \end{bmatrix} \tag{A.16}$$

Then equation (A.7) may be written:

$$\sum_{i=1}^{n} \overline{U}_i \overline{U}_i^T \overline{D}_i = \sum_{i=1}^{n} D_i e_i e_i^T - \frac{1}{\beta_n} V_n V_n^T \tag{A.17}$$

The elements of $\overline{D}$ and $\overline{U}$ may be determined from β, D and V by this relationship. Consider the matrix:

$$M = \sum_{i=1}^{n} \overline{U}_i \overline{U}_i^T \overline{D}_i - \sum_{i=1}^{n} D_i e_i e_i^T + \frac{1}{\beta_n} V_n V_n^T \tag{A.18}$$

The choices:

$$\overline{D}_n = D_n - \frac{V_{n,n}^2}{\beta_n} \tag{A.19a}$$

$$\overline{U}_{n,n} = 1 \tag{A.19b}$$

$$\overline{U}_{i,n} = -\frac{V_{n,n}}{\overline{D}_n \beta_n} V_{n,i}; \quad i=1,\ldots,n-1 \tag{A.19c}$$

will make the last row and column of M equal to zero. The matrix M may thus be written:

$$M = \left[\frac{V^2_{n,n}}{\overline{D}_n\beta^2_n} + \frac{1}{\beta_n}\right]V_{n-1}V^T_{n-1} \tag{A.20}$$

and from equations (A.5):

$$\frac{V^2_{n,n}}{\overline{D}_n\beta^2_n} + \frac{1}{\beta_n} = \frac{1}{\beta_{n-1}} \tag{A.21}$$

and:

$$\overline{D}_n = D_n\beta_{n-1}/\beta_n \tag{A.22}$$

By a similar argument each element of $\overline{D}$ may be calculated from D and β as:

$$\overline{D}_i = D_i\beta_{i-1}/\beta_i \tag{A.23}$$

The determinants may therefore be directly calculated from (A.23) and (A.5) as:

$$|\overline{D}| = \overline{D}_1\overline{D}_2\cdots\overline{D}_n = \frac{\beta_0}{\beta_1}D_1\frac{\beta_1}{\beta_2}D_2\cdots\frac{\beta_{n-1}}{\beta_n}D_n \tag{A.24}$$

$$= \frac{\beta_0}{\beta_n}|D| = \frac{\alpha}{(\alpha + x^TAx)}|A| \tag{A.25}$$

From (A.10) the proof is given:

$$|A'| = \frac{|A|}{\alpha^{n-1}(\alpha + x^TPx)} \tag{A.26}$$

rearranging and substituting (A.2) gives:

$$|\alpha A^{-1} + xx^T| = \frac{\alpha^{n-1}(\alpha + x^T A x)}{|A|} \tag{A.27}$$

□